REDUCED BASIS APPROXIMATION FOR NONLINEAR PARAMETRIZED EVOLUTION EQUATIONS BASED ON EMPIRICAL OPERATOR INTERPOLATION[*]

MARTIN DROHMANN[†], BERNARD HAASDONK[‡], AND MARIO OHLBERGER[†]

Abstract. We present a new approach to treat nonlinear operators in reduced basis approximations of parametrized evolution equations. Our approach is based on empirical interpolation of nonlinear differential operators and their Fréchet derivatives. Efficient offline/online decomposition is obtained for discrete operators that allow an efficient evaluation for a certain set of interpolation functionals. An a posteriori error estimate for the resulting reduced basis method is derived and analyzed numerically. We introduce a new algorithm, the PODEI-greedy algorithm, which constructs the reduced basis spaces for the empirical interpolation and for the numerical scheme in a synchronised way. The approach is applied to nonlinear parabolic and hyperbolic equations based on explicit or implicit finite volume discretizations. We show that the resulting reduced scheme is able to capture the evolution of both smooth and discontinuous solutions. In case of symmetries of the problem, the approach realizes an automatic and intuitive space–compression or even space–dimensionality reduction. We perform empirical investigations of the error convergence and run–times. In all cases we obtain a good run–time acceleration.

Key words. model reduction, parametrized evolution equations, reduced basis methods, empirical interpolation, a posteriori error estimation

AMS subject classifications. 65M08, 65M15, 65J15, 35L90, 35K90

1. Introduction.

The numerical solution of parametrized partial differential equations can be a very time-consuming task if many parameter constellations have to be simulated by high-resolution schemes. Such scenarios may occur in parameter studies, optimization, control, inverse problems or statistical analysis of a given parametrized problem. Reduced Basis (RB) methods allow to produce fast reduced models that are good surrogates for the original numerical scheme and allow parameter variations. These methods have gained increasing attention in recent years for stationary elliptic, instationary parabolic problems and various systems. In this contribution, we address the task of model reduction for parametrized evolution equations. These are problems which are characterized by a parameter vector $\boldsymbol{\mu} \in \mathcal{M}$ from some set of possible parameters $\mathcal{M} \subset \mathbb{R}^p$. For a given $\boldsymbol{\mu}$ the evolution problem consists of determining $u(x; t, \boldsymbol{\mu})$ on a bounded domain $\Omega \subset \mathbb{R}^d$ and finite time interval $t \in [0, T], T > 0$ such that

$$\partial_t u(t, \boldsymbol{\mu}) + \mathcal{L}(t, \boldsymbol{\mu}) \left[u(t, \boldsymbol{\mu}) \right] = 0, \quad u(0, \boldsymbol{\mu}) = u_0(\boldsymbol{\mu}), \tag{1.1}$$

and suitable boundary conditions are satisfied. Here $u_0(\boldsymbol{\mu})$ are the parameter dependent initial values and $\mathcal{L}(t, \boldsymbol{\mu})$ is a parameter dependent spatial differential operator. The initial value and the solution are supposed to have some spatial regularity $u_0(\boldsymbol{\mu}), u(t, \boldsymbol{\mu}) \in \mathcal{W} \subset L^2(\Omega)$.

[*] This work was supported by German Science Foundation (DFG) under the contract number OH 98/2-1.

[†] Institute of Numerical and Applied Mathematics, University of Münster, 48149 Münster, Germany (mdrohmann@uni-muenster.de). Questions, comments, or corrections to this document may be directed to that email address.

[‡] Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, 70569 Stuttgart, Germany. The second author was supported by the Baden-Württemberg Stiftung gGmbH and thanks the German Science Foundation (DFG) for financial support within the Cluster Of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

For the discrete solutions an H-dimensional discrete Hilbert space $\mathcal{W}_h \subset L^2(\Omega)$ with a suitable norm $\|\cdot\|_{\mathcal{W}_h}$ is assumed. Evolution schemes produce discrete solutions $u_h^k(\boldsymbol{\mu}) \in \mathcal{W}_h, k = 0, \ldots, K$ approximating $u(t^k, \boldsymbol{\mu})$ at the time instants $0 = t^0 < t^1 < \ldots < t^K = T$. These high-dimensional, detailed simulations are frequently expensive to compute due to the high space resolution and not suitable for use in multi-query settings, i.e. multiple simulation requests with varying parameters $\boldsymbol{\mu}$.

Reduced Basis (RB) methods are increasingly popular methods to solve such parametrized problems, aiming at reduced simulation schemes which approximate the detailed solutions $u_h^k(\boldsymbol{\mu})$ by efficiently computed reduced solutions $u_{\text{red}}^k(\boldsymbol{\mu}) \in \mathcal{W}_{\text{red}}$. Here $\mathcal{W}_{\text{red}} \subset \mathcal{W}_h \subset L^2(\Omega)$ is an N-dimensional reduced basis space with suitable reduced basis Φ_N. The latter is generated in a problem specific way based on snapshots. In particular, reduced basis methods have been applied successfully for various elliptic and parabolic problems, mainly based on finite element discretizations. For linear elliptic problems we refer to [22], linear parabolic equations are treated in [11]. Extensions to nonlinear equations [25, 9] or systems [23] have also been developed. For reduced methods based on finite volume discretizations we refer to [16].

In this contribution, we develop a new reduced basis framework for nonlinear partial differential equations. The approach is applicable to a large class of discretization schemes that are based on evaluations of discretized operators and their directional derivatives only. We exemplify our approach for finite volume schemes where Newton iterations are used for the solution of the resulting non-linear systems. The main ingredient that allows to guarantee this extent of generality is the empirical operator interpolation for discrete operators and their directional derivatives. A further trail of our extension is the derivation of a new a posteriori error estimate that can be used for error control and - in particular - to speed up the basis construction procedure in the offline phase of the reduced algorithm.

The idea of empirical interpolation was first proposed for data functions in [1] and used for reduced basis methods for elliptic and parabolic problems in [10, 20, 9, 2]. Preliminary results of the empirical operator interpolation for purely explicit operators were presented in [17, 7]. In the context of model reduction with proper orthogonal decomposition (POD)-methods, empirical interpolation for nonlinear finite difference matrices and their Jacobians was introduced in [4].

The structure of our paper is as follows: Section 2 introduces the empirical operator interpolation in full generality and explains its application on directional derivatives of discrete operators. As already mentioned, this is the key ingredient of our generalized reduced basis approach. In §3 a numerical scheme for evolution schemes is formulated including explicit and implicit contributions both depending nonlinearly on the solution. This scheme is the foundation for the reduced basis scheme presented in §4. We elaborate on the generation of the reduced basis space by the POD-GREEDY algorithm and on the nature and the costs of the offline/online-decomposition. A new a posteriori error estimator is derived in §5. In the experimental Section §6 we demonstrate the applicability of the resulting method for both smooth and discontinuous data subject to nonlinear convection and diffusion. Experimentally, we investigate the approximation properties and demonstrate the run–time gain compared to the full finite volume schemes. We conclude in §7.

2. Operator Approximation by Empirical Interpolation.

The reduced basis method requires the underlying numerical scheme to be written in a *separable* form allowing efficient decomposition of parameter dependent scalar functionals and precomputed parameter-independent operator parts. Therefore, we now introduce the empirical operator interpolation in order to approximate a - linear or nonlinear - parametrized discrete operator with a surrogate in a *separable* form suitable for efficient evaluations in a reduced basis scheme. The empirical interpolation method for operators as presented in this section gives rise to a reduced basis framework applicable for a very general class of numerical schemes for evolution equations. The operator based approach gives us the opportunity to rewrite numerical schemes by substituting the spatial operators with their empirical interpolants and to develop reduced schemes based on the high dimensional ones. For details we refer to §3 and §4. In §2.1 we also show that the directional derivative of a discrete operator can be interpolated efficiently.

Before we start with the description of the empirical operator interpolation, we introduce some notation used subsequently. We write $\mathcal{W}_h$ for a discrete function space defined on a closed subset $\Omega \subset \mathbb{R}^n$ with a non empty interior and a polygonal boundary. Following the notation of a finite element by P.G. Ciarlet [5], we define the set $\Sigma_h := \{\tau_i\}_{i=1}^H \subset \mathcal{W}_h'$ of linearly independent functionals, which are unisolvent on $\mathcal{W}_h$, i.e. there exist unique functions $\psi_i \in \mathcal{W}_h, i = 1, \dots, H$ which satisfy

$$\tau_j(\psi_i) = \delta_{ij}, \quad 1 \leq j \leq H.$$

The linear functionals $\tau_i, i = 1, \dots, H$ are called the *degrees of freedom* (DOFs) of the discrete function space $\mathcal{W}_h$ and the functions $\psi_i, i = 1, \dots, H$ are called *basis functions.* Note that these basis functions can e.g. be finite element, finite volume or discontinuous Galerkin basis functions on a numerical grid $\mathcal{T}_h \subset \Omega$.

We proceed to discretizations of (1.1) which is why from now on, $\mathcal{L}_h(t, \boldsymbol{\mu})$ always denotes a discretized (non-linear) operator acting on an H-dimensional discrete function space $\mathcal{W}_h$. In order to decompose the computations in an efficient online and an offline phase for high-dimensional data, the scheme must be formulated in a separable way, i.e. the discrete operators are written as a sum of products of efficiently computable parameter dependent functionals and high-dimensional basis functions that can be precomputed during the offline phase. Hence, we approximate the discrete operators by a separable interpolant $\mathcal{I}_M[\mathcal{L}_h(t, \boldsymbol{\mu})]$ of the form

$$\mathcal{I}_M \left[\mathcal{L}_h(t, \boldsymbol{\mu})\right] [v_h] := \sum_{m=1}^{M} \tau_m^{\mathrm{EI}} \left(\mathcal{L}_h(t, \boldsymbol{\mu}) \left[v_h\right]\right) \xi_m \approx \mathcal{L}_h(t, \boldsymbol{\mu}) \left[v_h\right] \tag{2.1}$$

for all $v_h \in \mathcal{W}_h$ with a parameter independent but space dependent *collateral reduced basis* $\boldsymbol{\xi}_M := \{\xi_m\}_{m=1}^M \subset \mathcal{W}_H$ and functionals $\tau_m^{\mathrm{EI}} : \mathcal{W}_h \to \mathbb{R}$ which must be computable with complexity independent of H. The sum is assumed to contain few terms, i.e. $M \ll H$. Such a separable approximation is obviously fully specified by defining the basis functions ξ_m, and the functionals $\tau_m^{\mathrm{EI}}, m = 1, \dots, M$. One can think of many reasonable choices for basis functions and corresponding coefficient functionals, but we focus on a specification resulting in the *empirical operator interpolation.*

2.1. Empirical Operator Interpolation. In this section, we adapt the empirical interpolation method introduced for functions in [1] for discretized operators. Firstly, we specify the generation of the collateral reduced basis space $\mathcal{W}_M := \langle \boldsymbol{\xi}_M \rangle$ and the coefficient functionals. Secondly, we show that the functionals can be computed efficiently, i.e. with complexity independent of the discrete function space dimension H. As a further extension, we show that the Fréchet derivative of a discrete operator can be efficiently approximated with the same collateral reduced basis as the operator itself.

Algorithm 2.1 Abstract algorithm for greedy basis generation

```
X-GREEDY($\mathcal{M}_{\text{train}}, \varepsilon_{\text{tol}}, \Upsilon_{\max}$ )
– Initialize reduced basis of dimension $\Upsilon_0$:
    $\mathcal{D}_{\Upsilon_0} \leftarrow$ X-INITBASIS()
    $\Upsilon \leftarrow \Upsilon_0$
repeat
    – Find parameter and time instance of worst approximated snapshot:
        $(\boldsymbol{\mu}_{\max}, t_{\max}) \leftarrow \arg\max_{(\boldsymbol{\mu},t)\in\mathcal{M}_{\text{train}}\times[t^0,\dots,t^K]}$ X-ERRORESTIMATE($\mathcal{D}_\Upsilon, \boldsymbol{\mu}, t$)
    – Extend reduced basis by snapshot:
        $\mathcal{D}_{\Upsilon+\gamma} \leftarrow$ X-EXTENDBASIS($\mathcal{D}_\Upsilon, \boldsymbol{\mu}_{\max}, t_{\max}$)
        $\Upsilon \leftarrow \Upsilon + \gamma$
until $\max_{(\boldsymbol{\mu},t)\in\mathcal{M}_{\text{train}}\times[t^0,\dots,t^K]}$ X-ERRORESTIMATE($\mathcal{D}_\Upsilon, \boldsymbol{\mu}, t$) $\leq \varepsilon_{\text{tol}}$ or $\Upsilon > \Upsilon_{\max}$
return reduced basis: $\mathcal{D}_\Upsilon$
```

Collateral basis generation. The method can briefly be expressed based on a set of interpolation DOFs $\Sigma_M := \{\tau_m^{EI}\}_{m=1}^M \subset \Sigma_h$ and a corresponding interpolation basis $\boldsymbol{\xi}_M$, i.e. $\tau_{m'}^{EI}[\xi_m] = \delta_{m,m'}$ for $1 \leq m, m' \leq M$. The generation process for these components works similarly to the algorithms described in the original empirical interpolation paper [1] in which point evaluations in so-called "magic points" are used as interpolation DOFs after the basis functions were selected.

The main idea is a greedy algorithm which iteratively enhances the reduced space with a new basis function. The selection of these basis functions is controlled by an error estimate which is minimized over a finite set of parameters $\mathcal{M}_{\text{train}} \subset \mathcal{M}$ and time instances $[t^0, \cdots, t^K]$. Algorithm 2.1 describes this strategy in an abstract way. We want to use this algorithm several times throughout this paper by specifying the methods

(i) X-INITBASIS() initializing the reduced basis,

(ii) X-ERRORESTIMATE() estimating the error between high dimensional and reduced snapshots and

(iii) X-EXTENDBASIS() adding solution snapshot to the reduced basis space. The number of added solution snapshots γ can be greater than 1.

The specialization of these methods for generating the collateral reduced basis and interpolation DOFs needed for the empirical interpolation of a specific discrete operator $\mathcal{L}_h(t, \boldsymbol{\mu}) : \mathcal{W}_h \to \mathcal{W}_h$ are given in Algorithm 2.2. Here, the greedy algorithm is used in the following way: The reduced basis data $\mathcal{D}_\Upsilon := (\mathbf{Q}_\Upsilon, \Sigma_\Upsilon)$ comprises the interpolation DOFs $\Sigma_\Upsilon := \left\{\tau_i^{EI}\right\}_{i=1}^{\Upsilon}$ and the collateral reduced basis functions $\mathbf{Q}_\Upsilon := \{q_i\}_{i=1}^{\Upsilon}$. The initial basis shall be empty and in each extension step, the interpolation error is used as an indicator for basis function selection. In the following, we denote the dimension of the final collateral reduced basis data with the upper case letter M. The algorithm defined by Algorithm 2.1 and specialized by the methods from Algorithm 2.2 will be named EI-GREEDY in the following.

REMARK 2.1. *The nodal basis $\boldsymbol{\xi}_M$ introduced in equation* (2.1) *is constructed from a basis $\mathbf{Q}_M := \{q_m\}_{m=1}^M$ made out of basis functions with a different structure, such that $\tau_m^{EI}[q_m] = 1$ and $\tau_{m'}^{EI}[q_m] = 0$ for all $m' > m$. Unlike the nodal basis, $\mathbf{Q}_M$ is constructed iteratively, such that $\mathbf{Q}_{M-1} \subset \mathbf{Q}_M$ and the basis functions' maximum norm is bound by one $\|q_m\|_{L^\infty(\Omega)} \leq 1$. The nodal basis $\boldsymbol{\xi}_M$ allows a simpler exposition of the functionals, and can be efficiently constructed from $\mathbf{Q}_M$, as the column matrix of basis vectors $q_m, m = 1, \dots, M$ has lower–triangular shape. In [1, 10] it is shown*

Algorithm 2.2 Methods for collateral reduced basis generation EI-greedy

```
EI-InitBasis()
  return empty initial basis: $\mathcal{D}_0 \leftarrow \{\}$

EI-ErrorEstimate($\left(\mathbf{Q}_M, \Sigma_M\right), \boldsymbol{\mu}, t^k$)
  – Compute exact operator evaluation
        $v_h \leftarrow \mathcal{L}_h(t^k, \boldsymbol{\mu})[u_h^k(\boldsymbol{\mu})]$
  – Compute interpolation coefficients $\boldsymbol{\sigma}^M(v_h) := \left(\sigma_j^M(v_h)\right)_{j=1}^M \in \mathbb{R}^M$
  by solving the linear equation system

        $\sum_{j=1}^M \sigma_j^M(v_h)\tau_i^{EI}\left[q_j\right] = \tau_i^{EI}\left[v_h\right], \quad i = 1, \dots, M$        (2.3)

  return approximation error: $\left\| v_h - \sum_{j=1}^M \sigma_j^M(v_h) q_j \right\|_{\mathcal{W}_h}$

EI-ExtendBasis($\left(\mathbf{Q}_M, \Sigma_M\right), \boldsymbol{\mu}, t^k$)
  – Compute exact operator evaluation
        $v_h \leftarrow \mathcal{L}_h(t^k, \boldsymbol{\mu})[u_h^k(\boldsymbol{\mu})]$
        and interpolation coefficients $\boldsymbol{\sigma}^M(v_h) := \left(\sigma_j^M(v_h)\right)_{j=1}^M \in \mathbb{R}^M$ from (2.3).
  – Compute the residual between $v_h$ and its current interpolant.
        $r_M \leftarrow v_h - \sum_{j=1}^M \sigma_j^M(v_h) q_j$
  – Find interpolation DOF maximizing the residual.
        $\tau_{M+1}^{EI} \leftarrow \arg\sup_{\tau \in \Sigma_h} |\tau(r_M)|$
  – Normalize to obtain a new collateral reduced basis function.
        $q_{M+1} \leftarrow (\tau_{M+1}^{EI}(r_M))^{-1} \cdot r_M$
  return extended basis data: $\mathcal{D}_{M+1} \leftarrow \left(\{q_m\}_{m=1}^{M+1}, \{\tau_m^{EI}\}_{m=1}^{M+1}\right)$
```

that the maximum norms of the nodal base functions can grow exponentially and that in the worst, but very unlikely case, the Lebesgue constant defined as

$$\Lambda_M := \sup_{x \in \Omega} \sum_{m=1}^M |\xi_m(x)|. \tag{2.2}$$

can reach its maximum of $2^M - 1$*. The Lebesgue constant correlates to the maximum ratio between the empirical interpolation of an operator evaluation and the operator evaluation's best approximation in the collateral reduced basis space. Therefore, an exponentially growing Lebesgue constant prevents the collateral reduced basis constructions from terminating as the targeted empirical interpolation error cannot be reduced. Nevertheless, it is suggested practice to assess the quality of the collateral reduced basis space by computing the growth of its Lebesgue constant.*

Due to this expected growth of the Lebesgue constant, we use the basis $\mathbf{Q}_M$ *in the implementation, but keep* $\boldsymbol{\xi}_M$ *for simpler exposition in the following paragraphs.*

Remark 2.2. *We mention, that the loop over the training set* $\mathcal{M}_{train}$ *which is necessary to find the worst approximation parameters in Algorithm 2.1 can be executed in parallel with hardly any communication costs. Here, only the scalar results of* X-ErrorEstimate *need to be communicated, such that the offline computation time can be extremely improved by use of parallel hardware.*

Evaluation of interpolation functionals $\boldsymbol{\tau}_{\mathbf{m}}^{\mathbf{EI}} \circ \mathcal{L}_{\mathbf{h}}(\mathbf{t}, \boldsymbol{\mu})$. An efficient evaluation of the functionals $\tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[v_h\right]\right)$ for every $\boldsymbol{\mu} \in \mathcal{M}$, $t \in [0, T_{\max}]$ and $v_h \in \{u_h(\boldsymbol{\mu}) | \boldsymbol{\mu} \in \mathcal{M}\}$ requires it to depend on few basis functions only. This fact inspires the following definition.

Definition 2.3 (H-independent DOF dependence). *A discrete operator $\mathcal{L}_h(t,\boldsymbol{\mu}) : \mathcal{W}_h \to \mathcal{W}_h$ fulfills an H-*independent DOF dependence*, if there exists a constant $C \ll H$ independent of H such that for all $\tau \in \Sigma_h$ a restriction operator $\mathcal{R}_\tau^C : \mathcal{W}_h \to \mathcal{W}_h, v_h = \sum_{i=1}^H \tau_i(v_h)\psi_i \mapsto \sum_{j \in I_\tau} \tau_j(v_h)\psi_j$ exists that restricts the operator argument to $|I_\tau| \leq C$ degrees of freedom and the equation*

$$\tau\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[v_h\right]\right) = \tau\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[\mathcal{R}_\tau^C\left[v_h\right]\right]\right)$$

still holds for all $v_h \in \mathcal{W}_h$.

Remark 2.4. *In particular, finite element or finite volume operators fulfill the H-independent DOF dependence, as a point evaluation of an operator application only requires data of the argument on neighbouring grid cells together with geometric information of this subgrid.*

Assuming this H-independence condition for a parametrized discrete operator, its empirical interpolant can be evaluated efficiently, i.e. independent of the dimension H. This result is summarized in the following corollary.

Corollary 2.5. *Let for each $t \in [0, T_{\max}]$ and $\boldsymbol{\mu} \in \mathcal{M}$ the discrete operators $\mathcal{L}_h(t,\boldsymbol{\mu}) : \mathcal{W}_h \to \mathcal{W}_h$ fulfill the H-*independent DOF dependence *and let Σ_M and $\boldsymbol{\xi}_M$ be determined for this operator by Algorithm 2.1 with specifications defined by Algorithm 2.2. Then, the empirical operator interpolation $\mathcal{I}_M$ defined by*

$$\mathcal{I}_M\left[\mathcal{L}_h(t,\boldsymbol{\mu})\right]\left[v_h\right] := \sum_{m=1}^M \tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[v_h\right]\right)\xi_m \tag{2.4}$$

gives a separable approximation of $\mathcal{L}_h(t,\boldsymbol{\mu})$ depending on at most CM degrees of freedom for each evaluation.

Ignoring the parameter independent collateral basis functions, an evaluation of the empirical interpolant has a complexity independent of the dimension H. This fact holds during a reduced basis simulation, because then all contributions depending on the collateral reduced basis have already been precomputed and reduced to vectors or matrices with dimensions independent of H as explained in §3 and §4.

Evaluation of functionals for the Fréchet derivative $\mathbf{D}\mathcal{L}_h(t,\boldsymbol{\mu})|_{u_h}[v_h]$. Many solvers for numerical approximations of nonlinear partial differential equations use the Newton method to resolve the nonlinearities in the equation and therefore depend on derivatives of discrete operators. It is easy to observe that the Fréchet derivative can also be applied to the empirical interpolant of an operator $\mathcal{L}_h(t,\boldsymbol{\mu})$ as

$$\mathbf{D}\left(\mathcal{I}_M\left[\mathcal{L}_h(t,\boldsymbol{\mu})|_{u_h}\right]\right)[\cdot] = \sum_{m=1}^M \mathbf{D}\left(\tau_m^{EI} \circ \mathcal{L}_h(t,\boldsymbol{\mu})|_{u_h}\right)[\cdot]\,\xi_m. \tag{2.5}$$

For an efficient usage of such an interpolation in a reduced scheme, it suffices to show that the functionals $\mathbf{D}\left(\tau_m^{EI} \circ \mathcal{L}_h(t,\boldsymbol{\mu})|_{u_h}\right)$ can be evaluated efficiently: With the chain rule for Fréchet derivatives and the existence of the derivatives w.r.t. the degrees of freedom, we obtain

$$\mathbf{D}\left(\tau_m^{EI} \circ \mathcal{L}_h(t,\boldsymbol{\mu})|_{u_h}\right)[v_h] = \sum_{i=1}^{H} \frac{\partial}{\partial \psi_i} \tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[u_h\right]\right)\tau_i(v_h) \tag{2.6}$$

$$= \sum_{i \in I_{\tau_m^{EI}}} \frac{\partial}{\partial \psi_i} \tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[u_h\right]\right)\tau_i(v_h). \tag{2.7}$$

The reduction in the number of summands holds true, as substituting the DOF $\tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[u_h\right]\right)$ by its restriction $\tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[\mathcal{R}^C_{\tau_m} u_h\right]\right)$ shows that most of the directional derivatives are zero. Each summand depends on one degree of freedom of the directional function v_h and at most C degrees of freedom of u_h summing up to an overall complexity of $C^2 M \ll H$. Again, this result with all its prerequisites is summarized in the following corollary.

Corollary 2.6. *Let the parametrized operators $\mathcal{L}_h(t,\boldsymbol{\mu})$ be as in Corollary 2.5 having a Fréchet derivative at the point $u_h \in \mathcal{W}_h$. Then, the Fréchet derivative of the empirical operator interpolation evaluated in direction $v_h \in \mathcal{W}_h$ is a separable approximation of $\mathbf{D}\mathcal{L}_h(t,\boldsymbol{\mu})|_{u_h}[v_h]$ with complexity independent of H for all $t \in [0, T_{\max}]$ and $\boldsymbol{\mu} \in \mathcal{M}$ if the derivatives $\frac{\partial}{\partial \psi_i}\tau_m^{EI}\left(\mathcal{L}_h(t,\boldsymbol{\mu})\left[u_h\right]\right)$ exist for all $i = 1,\ldots,H$ and $m = 1,\ldots,M$.*

Remark 2.7. *It is noteworthy, that the image $\cup_{\boldsymbol{\mu}\in\mathcal{M}}\mathrm{Im}\left(\mathcal{I}_M\left[\mathcal{L}_h(\boldsymbol{\mu})\right][\cdot]\right)$ is a subset of the convex hull of the original operators image $\mathbb{I} := \cup_{\boldsymbol{\mu}\in\mathcal{M}}\mathrm{Im}\left(\mathcal{L}_h(\boldsymbol{\mu})[\cdot]\right)$. Therefore, a property which applies to all functions of this convex hull $\mathrm{conv}(\mathbb{I})$ is preserved by the empirical operator interpolation. An example is the global conservation property stating that discrete functions $v_h \in \mathbb{I}$ have zero mean $\int_\Omega v_h = 0$. If such a property holds on a local part of the underlying domain space Ω or a subset of the parameter space $\mathcal{M}$ only, it can therefore be feasible to split the operator in two parts such that for one the desired property holds and is preserved by its empirical interpolant. For the global conservation property of finite volume operators, this result is discussed in Remark 3.4.*

The results of this section allow us to show in §4.4 how the empirical operator interpolation is utilized for the offline/online decomposition of the reduced basis scheme.

3. Evolution Scheme. In this section, we define an operator based numerical scheme which can be understood as a general formulation for standard discretizations of parametrized evolution problems (1.1). As an example, in §3.1 we present how these operators can be specified for a finite volume scheme. Together with the empirical interpolation for discrete operators and their Fréchet derivatives, a reduced scheme will be formulated in §4.

In what follows, we assume a first order discretization in time, and split the space discretization operator into implicit and explicit contributions. Both operator parts may depend nonlinearly on the argument. The non-linear implicit part will be treated by Newton iterations. For clarity of exposition, we fix the time step size, but of course, it would be possible to choose it adaptively in each step.

Definition 3.1 (General parametrized evolution scheme). *Let $\mathcal{W}_h$ be an H-dimensional discrete function space with a basis $\{\psi_i\}_{i=1}^H$ and $t^k := k\Delta t, k = 0,\ldots,K$ be a sequence of $K+1$ strictly increasing time instances with a global time step size $\Delta t > 0$. Furthermore, there needs to exist a projection $\mathcal{P}_h : L^2(\Omega) \to \mathcal{W}_h$ onto the discrete function space, and we assume an arbitrary space discretization operator $\mathcal{L}_h := \mathcal{L}_I + \mathcal{L}_E$ decomposed in its implicit and explicit contributions $\mathcal{L}_I :=$*

$\mathcal{L}_I(t^k,\boldsymbol{\mu}), \mathcal{L}_E := \mathcal{L}_E(t^k,\boldsymbol{\mu}) : \mathcal{W}_h \to \mathcal{W}_h$. For each parameter $\boldsymbol{\mu} \in \mathcal{M}$ we define a numerical scheme for discrete solutions $u_h^k := u_h^k(\boldsymbol{\mu}) = \sum_{i=1}^H u_{h,i}^k \psi_i \in \mathcal{W}_h$ at time instances t^k for $k = 0, \dots, K$ by initial projection

$$u_h^0 = \mathcal{P}_h\left[u_0(\boldsymbol{\mu})\right], \tag{3.1}$$

and subsequently solving the equations

$$F\left[u_h^{k+1}\right] := (\mathrm{Id} + \Delta t \mathcal{L}_I)\left[u_h^{k+1}\right] - (\mathrm{Id} - \Delta t \mathcal{L}_E)\left[u_h^k\right] = 0, \tag{3.2}$$

with the Newton-Raphson method. In each Newton step, we solve for the defect $\delta_h^{k+1,\nu+1}$ in

$$\mathbf{D}F|_{u_h^{k+1,\nu}}\left[\delta_h^{k+1,\nu+1}\right] = -F\left[u_h^{k+1,\nu}\right], \tag{3.3}$$

where $u_h^{k+1,0} := u_h^k$ and $u_h^{k+1,\nu+1} := u_h^{k+1,\nu} + \delta_h^{k+1,\nu+1}$ define the updates in each Newton step, and the solution at time instance t^k is given by $u_h^{k+1} := u_h^{k+1,\nu_{\max}^k}$. Here, the last Newton step index $\nu_{\max}^k$ equals the smallest integer ν satisfying the inequality

$$\left\|R_{h,New}^k\right\|_{\mathcal{W}_h} \leq \varepsilon^{New} \tag{3.4}$$

for the Newton residual

$$R_{h,New}^k := u_h^{k+1} - u_h^k + \Delta t\left(\mathcal{L}_I\left[u_h^{k+1}\right] + \mathcal{L}_E\left[u_h^k\right]\right) \tag{3.5}$$

and a predefined residual error bound $\varepsilon^{New} > 0$.

Note that, if $\mathcal{L}_I$ is linear, a single Newton-step is sufficient and in case $\mathcal{L}_I$ is zero, we obtain a purely explicit scheme. As a special case, the Crank–Nicolson scheme of second order is also covered.

3.1. Example: nonlinear finite volume scheme. As special instances of the general evolution equation (1.1), we consider the following scalar nonlinear convection–diffusion problem on a polygonal domain $\Omega \subset \mathbb{R}^2$ with the abbreviation $u = u(t;\boldsymbol{\mu})$ for a clearer exposition:

$$\partial_t u + \nabla \cdot (\mathbf{v}(u;\boldsymbol{\mu})u) - \nabla \cdot (d(u;\boldsymbol{\mu})\nabla u) = 0 \qquad \text{in } \Omega \times [0, T_{\max}] \tag{3.6}$$

with suitable parametrized functions $\mathbf{v}(\cdot;\boldsymbol{\mu}) \in C(\mathbb{R},\mathbb{R}^d)$ and $d(\cdot;\boldsymbol{\mu}) \in C(\mathbb{R},\mathbb{R}_0^+)$

$$u(0;\boldsymbol{\mu}) = u_0(\boldsymbol{\mu}) \qquad \text{in } \Omega \times \{0\}, \tag{3.7}$$

$$u(\boldsymbol{\mu}) = u_{\mathrm{dir}}(\boldsymbol{\mu}) \qquad \text{on } \Gamma_{\mathrm{dir}} \times [0, T_{\max}], \tag{3.8}$$

$$(\mathbf{v}(u;\boldsymbol{\mu})u - d(u;\boldsymbol{\mu})\nabla u) \cdot \mathbf{n} = u_{\mathrm{neu}}(\boldsymbol{\mu}) \qquad \text{on } \Gamma_{\mathrm{neu}} \times [0, T_{\max}] \tag{3.9}$$

and cyclical boundary conditions on the remaining boundary $\partial\Omega\backslash(\Gamma_{\mathrm{dir}} \cup \Gamma_{\mathrm{neu}})$. Here, $\mathbf{n}$ denotes the outer normal on the boundary. Note, that we also allow $d \equiv 0$.

We denote $\mathcal{W}$ as the exact solution space with respect to the space variable that can be chosen e.g. as $L^\infty(\Omega) \cap BV(\Omega) \subset L^2(\Omega)$. We obtain unique entropy solutions in $L^\infty(0, T_{\max}; \mathcal{W})$ if the data and boundary functions fulfill adequate regularity conditions. For discussion on well-posedness, uniqueness and existence of entropy solutions, see e.g. [3, 18].

Numerical scheme. Before we formulate the numerical scheme, we must introduce some notations. Let $\mathcal{T} := \{e_i\}_{i=1}^H$ denote a numerical grid consisting of H disjoint polygonal elements forming a partition of the domain $\bar{\Omega} = \bigcup_{i=1}^H \bar{e}_i$. For each element $e_i, i = 1, \ldots, H$, we assume that there exist certain points x_i lying inside the element e_i, such that points in adjacent elements are perpendicular to the corresponding edges. The cell's edges are denoted by e_{ij} with $j \in \mathcal{N}_{\text{in}}(i) \cup \mathcal{N}_{\text{neu}}(i) \cup \mathcal{N}_{\text{dir}}(i)$. Here, the index set $\mathcal{N}_{\text{in}}(i)$ comprises cell indices of all elements adjacent to e_i and $\mathcal{N}_{\text{neu}}(i)$ and $\mathcal{N}_{\text{dir}}(i)$ are enumerations of edges on which a Neumann respectively a Dirichlet condition is imposed. On each edge e_{ij}, we denote the barycenter by x_{ij} and the outer unit normal by $\mathbf{n}_{ij}$.

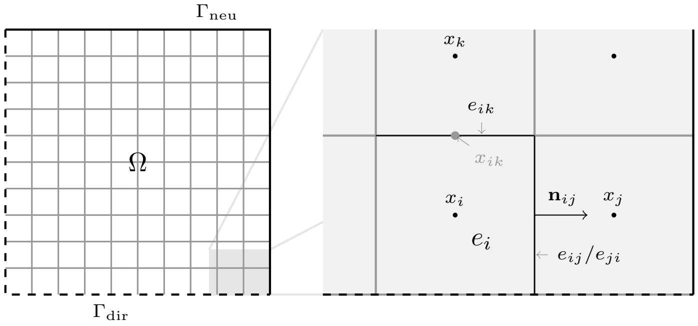


FIGURE 3.1. *Excerpt of a rectangular grid with notations used in this paper.*

The grid enables us to specify a discrete function space $\mathcal{W}_h := \text{span}\,\{\psi_i\}_{i=1}^H$ spanned by indicator functions $\psi_i := \chi_{e_i}$ piecewise constant on the grid cells. We denote the degrees of freedom of a function $u_h \in \mathcal{W}_h$ by $u_{h,i} = \tau_i(u_h) := u_h(x_i)$. For the time interval discretization, we choose the global time step size Δt small enough such that a CFL condition is fulfilled for all parameters $\boldsymbol{\mu} \in \mathcal{M}$.

The implicit and explicit space discretization operator need to model the diffusive respectively the convective dynamics of the underlying partial differential equations. Therefore we define

$$\mathcal{L}_I(\boldsymbol{\mu}) := \qquad \alpha\,\mathcal{L}_{\text{diff}}(\boldsymbol{\mu}) + \qquad \beta\,\mathcal{L}_{\text{conv}}(\boldsymbol{\mu}), \tag{3.10}$$

$$\mathcal{L}_E(\boldsymbol{\mu}) := (1-\alpha)\,\mathcal{L}_{\text{diff}}(\boldsymbol{\mu}) + (1-\beta)\,\mathcal{L}_{\text{conv}}(\boldsymbol{\mu}) \tag{3.11}$$

with constants $0 \leq \alpha, \beta \leq 1$ and finite volume operators $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{conv}}$ specified below. A judicious choice for the constants is $\alpha = 1$ and $\beta = 0$, because the greater stiffness of diffusion dynamics requires implicit discretizations, whereas for instationary problems, it is computationally more efficient to discretize convection terms explicitly. Note that the operators are constant in time, but the scheme applies to time–varying operators as well.

The main idea of the finite volume method is to compute cell–wise averages over the solutions and to substitute the occurring volume integrals containing divergence terms into surface integrals with the Gauss–Ostrogradsky theorem, such that e.g.

$$\nabla \cdot \varphi \approx \frac{1}{|e_i|} \int_{e_i} \nabla \cdot \varphi = \frac{1}{|e_i|} \int_{\partial e_i} \varphi \cdot \mathbf{n}. \tag{3.12}$$

For the diffusion operator, a finite difference approximation of the normal derivative gives us the DOF-wise definition

$$\begin{aligned}\left(\mathcal{L}_{\mathrm{diff}}(\boldsymbol{\mu})\left[u_h\right]\right)_i =& -\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{in}}(i)}\overline{d(u_h;\boldsymbol{\mu})_{ij}}\frac{u_{h,j}-u_{h,i}}{|x_j-x_i|}|e_{ij}|\\ &-\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{dir}}(i)}d(u_{\mathrm{dir}}(x_{ij};\boldsymbol{\mu});\boldsymbol{\mu})\frac{u_{\mathrm{dir}}(x_{ij};\boldsymbol{\mu})-u_{h,i}}{2|x_{ij}-x_i|}|e_{ij}|,\end{aligned} \tag{3.13}$$

where $|e_i|$ is the volume of the grid cell e_i and $\overline{d(u_h;\boldsymbol{\mu})_{ij}}$ computes a suitable mean on the edge e_{ij}, e.g. based on the harmonic mean $\frac{2d(u_j;\boldsymbol{\mu})d(u_i;\boldsymbol{\mu})}{d(u_j;\boldsymbol{\mu})+d(u_i;\boldsymbol{\mu})}$. In order to resolve the nonlinearity of the diffusion in a numerical scheme with the Newton-Raphson method, we also need the operator's directional derivative at a point u_h

$$\begin{aligned}\left(\mathbf{D}\mathcal{L}_{\mathrm{diff}}(\boldsymbol{\mu})|_{u_h}\left[v_h\right]\right)_i =& -\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{in}}(i)}\mathbf{D}\overline{d(\cdot;\boldsymbol{\mu})_{ij}}|_{u_h}\left[v_h\right]\frac{u_{h,j}-u_{h,i}}{|x_j-x_i|}|e_{ij}|\\ &+\overline{d(u_h;\boldsymbol{\mu})_{ij}}\frac{v_{h,j}-v_{h,i}}{|x_j-x_i|}|e_{ij}|\\ &-\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{dir}}(i)}d(u_{\mathrm{dir}}(x_{ij};\boldsymbol{\mu});\boldsymbol{\mu})\frac{-v_{h,i}}{2|x_{ij}-x_i|}|e_{ij}|.\end{aligned} \tag{3.14}$$

Likewise, we define the finite volume operator for the convection term by

$$\begin{aligned}\left(\mathcal{L}_{\mathrm{conv}}(\boldsymbol{\mu})\left[u_h\right]\right)_i =& \frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{in}}(i)}g_{ij}(u_{h,i},u_{h,j};\boldsymbol{\mu})\\ &+\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{dir}}(i)}g_{ij}(u_{h,i},u_{\mathrm{dir}}(x_{ij});\boldsymbol{\mu})\\ &+\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{neu}}(i)}\int_{e_{ij}}u_{\mathrm{neu}(\boldsymbol{\mu})}\end{aligned} \tag{3.15}$$

with Engquist-Osher flux functions g_{ij} leading to low numerical viscosity in this scheme. The flux functions can be expressed by setting $c_{ij}(u;\boldsymbol{\mu}) := \mathbf{n}_{ij}\mathbf{v}(u;\boldsymbol{\mu})u$ for all edges, defining

$$c^+_{ij}(u;\boldsymbol{\mu}) := c_{ij}(0;\boldsymbol{\mu}) + \int_0^u \max(c'_{ij}(s;\boldsymbol{\mu}),0)ds, \tag{3.16}$$

$$c^-_{ij}(u;\boldsymbol{\mu}) := \int_0^u \min(c'_{ij}(s;\boldsymbol{\mu}),0)ds \tag{3.17}$$

and choosing the flux as $g_{ij}(u,v;\boldsymbol{\mu}) := |e_{ij}|\left\{c^+_{ij}(u;\boldsymbol{\mu}) + c^-_{ij}(v;\boldsymbol{\mu})\right\}$, cf. [19]. The corresponding directional derivative w.r.t. v_h of the Engquist-Osher flux operator is given by

$$\begin{aligned}\left(\mathbf{D}\mathcal{L}_{\mathrm{conv}}(\boldsymbol{\mu})|_{u_h}\left[v_h\right]\right)_i =& \frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{in}}(i)}\partial_1 g_{ij}(u_{h,i},u_{h,j};\boldsymbol{\mu})v_{h,i} + \partial_2 g_{ij}(u_{h,i},u_{h,j};\boldsymbol{\mu})v_{h,j}\\ &+\frac{1}{|e_i|}\sum_{j\in\mathcal{N}_{\mathrm{dir}}(i)}\partial_1 g_{ij}(u_{h,i},u_{\mathrm{dir}}(x_{ij};\boldsymbol{\mu}))v_{h,i}.\end{aligned} \tag{3.18}$$

In order to complete the scheme, we can project the initial data onto the discrete function space via a cell averaging operator $\mathcal{P}_h : \mathcal{W} \to \mathcal{W}_h$, DOF-wise defined by $\tau_i \left(\mathcal{P}_h \left[v_h\right]\right) := \frac{1}{|e_i|} \int_{e_i} v_h$ and obtain a specification of the generalized numerical scheme from Definition 3.1.

REMARK 3.2 (Restriction to interpolation DOFs). *From the DOF-wise definitions of the operators* (3.13) *and* (3.15)*, it follows that a constant number of flops dependent on the maximum number of cell neighbours suffices to numerically compute a single degree of freedom from the operator evaluation result. Therefore, the finite volume operators fulfill the H-independent DOF dependence condition and are suitable for empirical interpolation with the constant C bounded by one plus the maximum number of edges of an element.*

A crucial property of finite volume operators is its local conservation property assuring that everything flowing out of a cell, flows into a neighbouring one. For an arbitrary finite volume (update) operator in the flux formulation

$$\left(\mathcal{L}_h \left[u_h\right]\right)_i = \sum_{j \in \mathcal{N}(i)} g_{ij}(u_h). \tag{3.19}$$

the local conservation property holds, iff

$$g_{ij}(u_h) = -g_{ji}(u_h). \tag{3.20}$$

In order to discuss that the conservation property is preserved by the empirical interpolant of a parametrized finite volume operator, we assume the EI-GREEDY algorithm has selected the time step indices and parameters $\left\{(k_1^{EI}, \boldsymbol{\mu}_1^{\mathrm{EI}}), \dots, (k_M^{EI}, \boldsymbol{\mu}_M^{\mathrm{EI}})\right\}$ for the extension of the collateral reduced basis space, such that the basis functions are given by

$$q_m = c_m \left(\mathcal{L}_h \left[u_h^{k_m^{EI}}(\boldsymbol{\mu}_m^{EI})\right] - \mathcal{I}_{m-1}\left[\mathcal{L}_h\right]\left[u_h^{k_m^{EI}}(\boldsymbol{\mu}_m^{EI})\right]\right) \tag{3.21}$$

with normalization factors $c_m := \left((\tau_m^{EI}(r_{m-1}))\right)^{-1}$ (c.f. method EI-EXTENDBASIS from Algorithm 2.2). The following lemma proves the preservation of the local conservation property.

LEMMA 3.3. *If for all $\boldsymbol{\mu} \in \mathcal{M}$ the parametrized finite volume operator $\mathcal{L}_h(\boldsymbol{\mu}) : \mathcal{W}_h \to \mathcal{W}_h$ has a locally conservative flux $g(\boldsymbol{\mu}; \cdot)$, then the empirical interpolant $\mathcal{I}_M \left[\mathcal{L}_h(\boldsymbol{\mu})\right]$ can be evaluated by*

$$\left(\mathcal{I}_M\left[\mathcal{L}_h(\boldsymbol{\mu})\right]\left[v_h\right]\right)_i = \sum_{j \in \mathcal{N}(i)} g_{ij}(\boldsymbol{\mu}; v_h) \tag{3.22}$$

for all $v_h \in \mathcal{W}_h$ with a parametrized flux $g^{\mathcal{I}_M}(\boldsymbol{\mu}; \cdot)$ recursively defined by

$$\begin{aligned} g_{ij}^{\mathcal{I}_M}(\boldsymbol{\mu}; v_h) := \sigma_M(v_h) &\left(g_{ij}\left(\boldsymbol{\mu}_M^{EI}; u_h^{k_M^{EI}}(\boldsymbol{\mu}_M^{EI})\right) - g_{ij}^{\mathcal{I}_{M-1}}\left(\boldsymbol{\mu}_M^{EI}; u_h^{k_M^{EI}}(\boldsymbol{\mu}_M^{EI})\right)\right) \\ &+ g_{ij}^{\mathcal{I}_{M-1}}(\boldsymbol{\mu}; v_h) \end{aligned} \tag{3.23}$$

for all $M \in \mathbb{N}_{>0}$ and $g_{ij}^{\mathcal{I}_0}(\boldsymbol{\mu}; \cdot) := 0$. It follows that this flux inherits the local conservation property from $g(\boldsymbol{\mu}; \cdot)$.

Proof. From equation (3.21), it follows by induction and with equation (3.19) that

$$(q_M)_i = c_M \cdot \sum_{j \in \mathcal{N}(i)} g_{ij}\left(\boldsymbol{\mu}_M^{EI}; u_h^{k_M^{EI}}(\boldsymbol{\mu}_M^{EI})\right) - g_{ij}^{\mathcal{I}_{M-1}}\left(\boldsymbol{\mu}_M^{EI}; u_h^{k_M^{EI}}(\boldsymbol{\mu}_M^{EI})\right). \tag{3.24}$$

Then, from $\mathcal{I}_M \left[\mathcal{L}_h(\boldsymbol{\mu})\right][v_h] = \sum_{m=1}^{M} \sigma_m^M(v_h) q_m$ follows

$$\left(\mathcal{I}_M \left[\mathcal{L}_h(\boldsymbol{\mu})\right][v_h]\right)_i = \sigma_M^M(v_h)\left(q_M\right)_i + \sum_{j \in \mathcal{N}(i)} g_{ij}^{\mathcal{I}_{M-1}}\left(v_h\right) \tag{3.25}$$

and such, after substituting (3.24) into (3.25),

$$g_{ij}^{\mathcal{I}_M}\left(v_h\right) = -g_{ji}^{\mathcal{I}_M}\left(v_h\right), \tag{3.26}$$

because $g(\boldsymbol{\mu};\cdot)$ and $g^{\mathcal{I}_{M-1}}(\boldsymbol{\mu};\cdot)$ are both conservative fluxes by assumption and by induction, respectively. The sum of conservative fluxes stays conservative. □

REMARK 3.4 (Global conservation under empirical interpolation). *As a corollary from the previous lemma or from Remark 2.7, it follows that global conservation, i.e.*

$$\int_\Omega \mathcal{L}_h[v_h] = 0 \qquad \textit{for all } v_h \in \mathcal{W}_h, \tag{3.27}$$

is also preserved under empirical interpolation. In case of trivial boundary conditions, the operators (3.13) *and* (3.15) *are for all* $\boldsymbol{\mu} \in \mathcal{M}$ *conservative in this sense. For non–trivial boundary conditions, however, the operators can be split into one operator for the fluxes over domain boundaries and one for the fluxes over inner grid cell interfaces as described in Remark 2.7.*

Note, that the interpolation procedure and the reduced scheme are identically applicable to other evolution problems, discrete function spaces and discretization operators, e.g. finite element or discontinuous Galerkin methods. Hence, for the following development of the reduced basis method, we will express the numerical scheme in terms of the more general notions from Definition 3.1.

4. Reduced Basis Method.

The key ingredient for a reduced basis scheme is the availability of a suitable low dimensional reduced basis space $\mathcal{W}_{\text{red}}$. In this section, we give a short review of an algorithm for reduced basis generation. We refer to the article [22] for a more detailed presentation on this topic. In the following §4.3 and §4.4, we introduce a reduced basis scheme for the general evolution problem from Definition 3.1 and comment on its suitability for an efficient offline/online decomposition.

In the experiments, we apply the POD-GREEDY reduced basis construction method as presented in [16] and a new more sophisticated alogorithm (PODEI-GREEDY) extending the reduced basis and the collateral reduced basis spaces in a synchronised way. Both algorithm are used with adaptive training parameter set extension introduced in [13]. Different approaches that combine adaptive parameter selection with the generation of multiple reduced basis spaces for different parameter sets have recently been given in [8, 13]. Convergence rate statements haven been obtained for an idealized version of the algorithm [12] as a theoretical foundation. All these methods are inherently accumulative and snapshot-based, like the collateral reduced basis generation schemes described above. An initially small (or empty) basis is iteratively enriched based on solutions $u_h^{k_i}(\boldsymbol{\mu}_i)$ for certain time steps k_i and parameters $\boldsymbol{\mu}_i$.

4.1. EI-greedy + POD-greedy basis generation.

In order to generate a suitable reduced basis space approximating the manifold of “interesting” solution snapshots $\left\{u_h^k(\boldsymbol{\mu}) \in \mathcal{W}_h | \boldsymbol{\mu} \in \mathcal{M}, k = 0, \ldots, K\right\}$, we start with a small initial reduced basis and then perform the greedy search Algorithm 2.1 with the methods described in Algorithm 4.1. This algorithm will be named POD-GREEDY in the following.

Algorithm 4.1 Methods for the POD-GREEDY algorithm

```
POD-InitBasis()
  return initial reduced basis functions: $\{\varphi_n\}_{n=1}^{N_0}$

POD-ErrorEstimate($\{\varphi_n\}_{n=1}^{N}, \boldsymbol{\mu}, t^k$)
  return error estimate: $\eta_{N,M}^k(\boldsymbol{\mu}) \geq \left\| u_{\mathrm{red}}^k(\boldsymbol{\mu}) - u_h^k(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$

POD-ExtendBasis($\{\varphi_n\}_{n=1}^{N}, \boldsymbol{\mu}_{\max}, t$)
  – Compute trajectory $\left\{ u_h^k(\boldsymbol{\mu}_{\max}) \right\}_{k=0}^{K}$.
  – Compute new basis function with Galerkin projection $\mathcal{P}_{\mathrm{red}}$ projecting
    onto span $\{\varphi_n\}_{n=1}^{N}$
        $\varphi_{N+1} \leftarrow \mathrm{POD}\left( \left\{ u_h^k(\boldsymbol{\mu}_{\max}) - \mathcal{P}_{\mathrm{red}}\left[ u_h^k(\boldsymbol{\mu}_{\max}) \right] \right\}_{k=0}^{K} \right)$
  return extended reduced basis: $\{\varphi_n\}_{n=1}^{N+1}$
```

We assume that the quality of reduced simulation trajectories $\left\{ u_{\mathrm{red}}^k(\boldsymbol{\mu}) \right\}_{k=0}^{K}$ which are obtained by the reduced numerical scheme introduced in §4.3, can be assessed by a posteriori error bounds. These error estimates denoted by $\eta_{N,M}^k : \mathcal{M} \to \mathbb{R}$ bound the reduction error $\left\| u_{\mathrm{red}}^k(\boldsymbol{\mu}) - u_h^k(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$ for every $k = 0, \ldots, K$ and different dimensions N and M of the reduced basis respectively the collateral reduced basis space. Of course, it is also possible to use the exact error directly. However, a direct evaluation of the error depends on a large number of inefficient computations. Therefore, more efficient a posteriori estimators allowing to deal with a large number of training samples, are preferable. This is necessary in order to assure the reduced basis approximation can be good enough for all parameters. An example for an error estimator separable into parameter independent offline and efficient online computations is given in §5.

For an extension step of the reduced basis space the entire solution trajectory $\left\{ u_h^k(\boldsymbol{\mu}_{\max}) \right\}_{k=0}^{K}$ to the worst approximated parameter $\mu_{\max}$ is considered. A proper orthogonal decomposition (POD) on this trajectory's projection error is applied and only the dominant modes are used for the basis extension. In the extension algorithm used for the experiments and presented in Algorithm 4.1, we are using only the single most dominant mode.

The initialization of the reduced basis generation is usually realized by adding initial data projections $\mathcal{P}_h\left[u_0(\boldsymbol{\mu}) \right]$. Assuming an affine parameter dependence of the initial data function u_0, it is possible to assure that these projections lie in the reduced basis space when it includes all parameter independent contributions.

It is important to note, that error estimates computed by POD-ErrorEstimate() include reduced computations which depend on the empirical interpolation of nonlinear operators. Therefore, the empirical interpolation bases must be computed with Algorithms 2.1 + 2.2 for all discrete operators of the numerical scheme beforehand. This approach has some drawbacks:

(i) The empirical interpolation bases are generated such that an artificial interpolation error is reduced for which it is not clear, how it relates to the error estimates $\eta_{N,M}^k$ used in the POD-greedy algorithm. Therefore, it is impossible to determine a priori the optimal correlation between the reduced basis space and the collateral reduced basis space.

(ii) The empirical interpolation error estimation by EI-ErrorEstimate() depends on high dimensional computations for each parameter and time step tested.

This can be very inefficient for large parameter sets $\mathcal{M}_{\text{train}}$.

(iii) In our experiments the reduced basis generation can be improved if the training parameter set $\mathcal{M}_{\text{train}}$ is adapted during the basis generation. This allows to begin with a small parameter set and to reduce the computation time for the basis generation. The POD-GREEDY algorithm finds a good training parameters set, but this set is unknown at the stage when the collateral reduced basis is generated.

4.2. PODEI-greedy basis generation. In this section we will introduce another greedy algorithm for synchronised execution of "POD greedy and empirical interpolation basis generation" (PODEI). This algorithm is again based on Algorithm 2.1 but generates the reduced basis and the collateral reduced basis spaces in parallel and overcomes the drawbacks of the subsequent execution of the algorithms EI-GREEDY and POD-GREEDY. The methods for the PODEI-GREEDY algorithm are sketched in Algorithm 4.2. A similar approach of a synchronized generation of reduced basis spaces has recently been published in [24].

The proposed algorithm uses only the error estimates $\eta^k_{N,M}$ for a greedy search in the parameter samples $\mathcal{M}_{\text{train}} \subset \mathcal{M}$ and attempts to extend both reduced spaces in each step. In previous applications of the empirical operator interpolation to nonlinear operators in reduced basis schemes [17, 7], it was observed that numerical schemes become unstable if the accuracy of the empirical interpolation is too bad with respect to the accuracy of the reduced basis space. In order to avoid this behaviour during the basis generation, we discard newly computed reduced basis functions if they increase the estimated error. This leads to an automatic control of the M-N correlation between the dimensions of the two basis spaces. A similar idea for this automatic control of the two basis sizes is presented in [24] where so–called *EIM plateaus* are identified on which the model reduction error is dominated by the projection error made by the projection on the reduced basis space. On these EIM plateaus, a further extension of empirical interpolation basis functions is useless.

It is noteworthy, that the initial collateral basis is generated by a full EI-GREEDY algorithm generating a small initial basis on a coarser parameter sampling set $\mathcal{M}^{\text{coarse}}_{\text{train}}$.

4.3. Reduced Basis Scheme. In this section we introduce a reduced basis scheme based on the formulation in [17, 15] for explicit discretizations of evolution problems. We extend the scheme by allowing nonlinear or non-separable implicit operator contributions. The basic idea for the reduced basis scheme is to replace the discrete evolution operators $\mathcal{L}_E$ and $\mathcal{L}_I$ from Definition 3.1 by their empirical interpolants $\mathcal{I}_M\left[\mathcal{L}_E\right]$ and $\mathcal{I}_M\left[\mathcal{L}_I\right]$ and applying an orthogonal projection of the numerical scheme onto the reduced basis space $\mathcal{W}_{\text{red}}$ with respect to the scalar product of $\mathcal{W}_h$. For this purpose, we introduce the corresponding orthogonal projection operator $\mathcal{P}_{\text{red}} : \mathcal{W}_h \to \mathcal{W}_{\text{red}}$ satisfying

$$\langle \mathcal{P}_{\text{red}}\left[u\right], \varphi\rangle = \langle u, \varphi\rangle \qquad \forall \varphi \in \mathcal{W}_{\text{red}}$$

and define reduced variants of the discrete operators

$$\mathcal{L}_{\text{red},E} := \mathcal{P}_{\text{red}} \circ \mathcal{I}_M \circ \mathcal{L}_E \quad \text{and} \quad \mathcal{L}_{\text{red},I} := \mathcal{P}_{\text{red}} \circ \mathcal{I}_M \circ \mathcal{L}_I. \tag{4.1}$$

For all $\boldsymbol{\mu} \in \mathcal{M}$ we obtain trajectories $\left\{u^k_{\text{red}}(\boldsymbol{\mu})\right\}^K_{k=0}$ with snapshots $u^k_{\text{red}}(\boldsymbol{\mu}) \in \mathcal{W}_{\text{red}}$ for $k = 0, \ldots, K$ analogously to the evolution scheme described in Definition 3.1. The reduced initial data is given by projection of the initial data

$$u^0_{\text{red}} := u^0_{\text{red}}(\boldsymbol{\mu}) = \mathcal{P}_{\text{red}}\left[\mathcal{P}_h\left[u_0(\boldsymbol{\mu})\right]\right]. \tag{4.2}$$

Algorithm 4.2 Methods for the PODEI-GREEDY algorithm

```
PODEI-InitBasis()
  – Generate small empirical interpolation basis:
        $(\mathbf{Q}_{M_{\text{small}}}, \Sigma_{M_{\text{small}}}) \leftarrow$ EI-greedy($\mathcal{M}_{\text{train}}^{\text{coarse}}, \varepsilon_{\text{tol,small}}, M_{\text{small}}$)
  – Compute initial reduced basis:
        $\{\varphi_n\}_{n=0}^{N_0} \leftarrow$ POD-InitBasis()
  return initial bases data: $\mathcal{D}_1 \leftarrow \{\varphi_n\}_{n=1}^{N_0} \cup (\mathbf{Q}_{M_{\text{small}}}, \Sigma_{M_{\text{small}}})$

PODEI-ErrorEstimate($\mathcal{D}_\Upsilon, \boldsymbol{\mu}, t^k$)
  return reduced basis error estimate: $\eta_{N,M}^k(\boldsymbol{\mu})$

PODEI-ExtendBasis($\mathcal{D}_\Upsilon, \boldsymbol{\mu}_{\max}, t^k$)
  Reduced data $\mathcal{D}_\Upsilon$ comprises $\mathcal{D}_N^{RB} := \{\varphi_n\}_{n=1}^N$ and $\mathcal{D}_M^{EI} := (\mathbf{Q}_M, \Sigma_M)$
  – Extend EI basis: $\mathcal{D}_{M+1}^{\text{EI}} \leftarrow$ EI-ExtendBasis($\mathcal{D}_M^{\text{EI}}, \boldsymbol{\mu}_{\max}, t^k$)
  – Extend RB basis: $\mathcal{D}_{N+1}^{\text{RB}} \leftarrow$ POD-ExtendBasis($\mathcal{D}_N^{\text{RB}}, \boldsymbol{\mu}_{\max}, t^k$)
  – Discard extended RB if error increases:
  if $\eta_{N-1,M-1}^k(\boldsymbol{\mu}_{\max}) \leq \max_{(\boldsymbol{\mu},t)\in\mathcal{M}_{\text{train}}} \eta_{N,M}^k(\boldsymbol{\mu})$ then
      return extended basis data: $\mathcal{D}_{\Upsilon+1} \leftarrow (\mathcal{D}_N^{\text{RB}}, \mathcal{D}_{M+1}^{\text{EI}})$
  else
      return extended basis data: $\mathcal{D}_{\Upsilon+1} \leftarrow (\mathcal{D}_{N+1}^{\text{RB}}, \mathcal{D}_{M+1}^{\text{EI}})$
  end if
```

Then, for each $k = 0, \dots, K-1$ Newton step solutions are computed by finding for $\nu = 0, \dots, \nu_{\max}^k(\boldsymbol{\mu})$ defects $\delta_{\text{red}}^{k+1,\nu} := \delta_{\text{red}}^{k+1,\nu}(\boldsymbol{\mu})$ solving

$$\begin{aligned}\left(\text{Id} + \Delta t \mathbf{D}\mathcal{L}_{\text{red},I}\big|_{u_{\text{red}}^{k+1,\nu}}\right)\left[\delta_{\text{red}}^{k+1,\nu+1}\right] &= -u_{\text{red}}^{k+1,\nu} + u_{\text{red}}^k \\ &\quad - \Delta t\left[\mathcal{L}_{\text{red},I}\left[u_{\text{red}}^{k+1,\nu}\right] + \mathcal{L}_{\text{red},E}\left[u_{\text{red}}^k\right]\right],\end{aligned} \tag{4.3}$$

with $u_{\text{red}}^{k+1,0} := u_{\text{red}}^k$, $u_{\text{red}}^{k+1,\nu+1} := u_{\text{red}}^{k+1,\nu} + \delta_{\text{red}}^{k+1,\nu+1}$ for $\nu = 1, \dots, \nu_{\max}^k - 1$, and finally assigning the reduced solution for the next time step by

$$u_{\text{red}}^{k+1} := u_{\text{red}}^{k+1,\nu_{\max}^k}. \tag{4.4}$$

Here, the final Newton iteration index ν_{max}^k is the smallest integer such that the norm of the residual defined as

$$R_{\text{red,New}}^k := u_{\text{red}}^{k+1} - u_{\text{red}}^k + \Delta t\left(\mathcal{L}_{\text{red},I}\left[u_{\text{red}}^{k+1}\right] + \mathcal{L}_{\text{red},E}\left[u_{\text{red}}^k\right]\right) \tag{4.5}$$

drops below a given tolerance ε^{New}.

REMARK 4.1. *If $\mathcal{L}_I$ is linear, a single Newton-step is sufficient. If $\mathcal{L}_I$ is zero, no Newton step at all is necessary and the numerical scheme is purely explicit. This case has already been discussed in [17]. Non-linear parabolic problems with finite element discretizations are also discussed in [11, 2] and [10]. Similar to the empirical operator interpolation based approach presented in this paper, those reduced basis methods make use of an empirical interpolation applied to specific data functions. In this case, the interpolation functionals $\{\tau_m^{EI}\}_{m=1}^M$ also define point evaluations at "magic points" for which the "H-independent DOF dependence" is trivially fulfilled with $C = 1$.*

For easier analysis of the computational complexity during offline and online phase, we translate the above sketched reduced scheme into a vector-valued formulation based on the few degrees of freedom of the reduced solution.

Definition 4.2 (Reduced basis scheme with empirical operator interpolation). *We assume a numerical scheme from Definition 3.1 with operators $\mathcal{L}_E$ and $\mathcal{L}_I$ fulfilling an H-independent DOF dependence. Hence, we can assume that an appropriate empirical interpolation operator $\mathcal{I}_M$ is defined by means of an empirical interpolation basis $\boldsymbol{\xi}_M$ and an enumerated subset of degrees of freedom $\Sigma_M := \left\{\tau_m^{EI}\right\}_{m=1}^M \subset \Sigma_h$. The collateral reduced basis space shall be the same for both the operators $\mathcal{L}_E$ and $\mathcal{L}_I$ (c.f. Remark 4.3). Furthermore, there must be an orthonormal reduced basis $\Phi_N := \{\varphi_n\}_{n=1}^N$ available that spans the reduced basis space $\mathcal{W}_{\mathrm{red}}$.*

We define the following scheme for sequentially expressing

(i) *the reduced solution $u_{\mathrm{red}}^k(\boldsymbol{\mu}) := \sum_{n=1}^N a_n^k(\boldsymbol{\mu})\varphi_n$,*

(ii) *intermediate Newton step solutions $u_{\mathrm{red}}^{k,\nu}(\boldsymbol{\mu}) := \sum_{n=1}^N a_n^{k,\nu}(\boldsymbol{\mu})\varphi_n$ and*

(iii) *Newton step defects $\delta_{\mathrm{red}}^{k,\nu}(\boldsymbol{\mu}) := \sum_{n=1}^N d_n^{k,\nu}(\boldsymbol{\mu})\varphi_n$*

by computing the coefficient vectors

$$\mathbf{a}^k := \mathbf{a}^k(\boldsymbol{\mu}) = \left(a_1^k(\boldsymbol{\mu}), \dots, a_N^k(\boldsymbol{\mu})\right)^T, \qquad \mathbf{a}^{k,\nu} := \mathbf{a}^{k,\nu}(\boldsymbol{\mu}) = \left(a_1^{k,\nu}(\boldsymbol{\mu}), \dots, a_N^{k,\nu}(\boldsymbol{\mu})\right)^T$$
$$\text{and } \mathbf{d}^{k,\nu} := \mathbf{d}^{k,\nu}(\boldsymbol{\mu}) = \left(d_1^{k,\nu}(\boldsymbol{\mu}), \dots, d_N^{k,\nu}(\boldsymbol{\mu})\right)^T$$

for $k = 0, \dots, K$ and $\nu = 0, \dots, \nu_{\max}^k(\boldsymbol{\mu})$:

The initial solution vector is obtained by projection onto the reduced basis space

$$\mathbf{a}^0 := \left(\left(\mathcal{P}_{\mathrm{red}}\left[\mathcal{P}_h[u_0(\boldsymbol{\mu})], \varphi_1\right]\right), \dots, \left(\mathcal{P}_{\mathrm{red}}\left[\mathcal{P}_h[u_0(\boldsymbol{\mu})], \varphi_N\right]\right)\right)^T. \tag{4.6}$$

Then, for each time index $k = 0, \dots, K-1$ we compute Newton iterations by finding defects $\mathbf{d}^{k+1,\nu+1}$ and residuals $\mathbf{r}^{k+1,\nu+1}$ solving for $\nu = 0, \dots, \nu_{\max}^k(\boldsymbol{\mu})-1$ the equations

$$\begin{aligned}\left(\mathrm{Id} + \Delta t \mathbf{C}\mathbf{l}_I'(t^k;\boldsymbol{\mu})\left[\mathbf{a}^{k+1,\nu}\right]\right)\left[\mathbf{d}^{k+1,\nu+1}\right] \\ = -\mathbf{a}^{k+1,\nu} + \mathbf{a}^{k+1,0} \\ - \Delta t\mathbf{C}\left(\mathbf{l}_I(t^k;\boldsymbol{\mu})\left[\mathbf{a}^{k+1,\nu}\right] + \mathbf{l}_E(t^k;\boldsymbol{\mu})\left[\mathbf{a}^{k+1,0}\right]\right),\end{aligned} \tag{4.7}$$

$$\begin{aligned}\mathbf{r}^{k+1,\nu+1}(\boldsymbol{\mu}) := \mathbf{a}^{k+1,\nu+1} - \mathbf{a}^{k+1,0} \\ + \Delta t\mathbf{C}\left(\mathbf{l}_I(t^k;\boldsymbol{\mu})\left[\mathbf{a}^{k+1,\nu+1}\right] + \mathbf{l}_E(t^k;\boldsymbol{\mu})\left[\mathbf{a}^{k+1,0}\right]\right)\end{aligned} \tag{4.8}$$

with updates

$$\begin{aligned}\mathbf{a}^{k+1,0} &:= \mathbf{a}^k, \\ \mathbf{a}^{k+1,\nu+1} &:= \mathbf{a}^{k+1,\nu} + \mathbf{d}^{k+1,\nu+1}, \\ \mathbf{a}^{k+1} &:= \mathbf{a}^{k+1,\nu_{\max}^k(\boldsymbol{\mu})}.\end{aligned} \tag{4.9}$$

The number of Newton steps $\nu_{\max}^k(\boldsymbol{\mu})$ at each time step is chosen as the smallest integer ν such that the residual norm drops below the specified tolerance for the Newton scheme, i.e. for which $\left(\left(\mathbf{r}^{k+1,\nu+1}(\boldsymbol{\mu})\right)^T \mathbf{M}\mathbf{r}^{k+1,\nu+1}(\boldsymbol{\mu})\right)^{\frac{1}{2}} < \varepsilon^{New}$ holds.

The utilized vectors and matrices are defined as

$$\left(\mathbf{M}\right)_{nn'} := \left\langle \varphi_n, \varphi_{n'} \right\rangle = \delta_{nn'}, \tag{4.10}$$

$$\left(\mathbf{C}\right)_{nm} := \left\langle \xi_m, \varphi_n \right\rangle, \tag{4.11}$$

$$\left(\mathbf{l}'_I\left(t^k,\boldsymbol{\mu}\right)\left[\mathbf{a}^{k+1,\nu}\right]\right)_{mn} := \sum_{i=1}^{H} \frac{\partial}{\partial \psi_i}\left(\tau_m^{EI} \circ \mathcal{L}_I(t^k,\boldsymbol{\mu})\right)\left[u_{\text{red}}^{k+1,\nu}\right]\tau_i\left(\varphi_n\right), \tag{4.12}$$

$$\left(\mathbf{l}_I\left(t^k,\boldsymbol{\mu}\right)\left[\mathbf{a}^{k+1,\nu}\right]\right)_m := \tau_m^{EI}\left(\mathcal{L}_I(t^k,\boldsymbol{\mu})\left[u_{\text{red}}^{k+1,\nu+1}\right]\right), \tag{4.13}$$

$$\left(\mathbf{l}_E\left(t^k,\boldsymbol{\mu}\right)\left[\mathbf{a}^{k}\right]\right)_m := \tau_m^{EI}\left(\mathcal{L}_E(t^k,\boldsymbol{\mu})\left[u_{\text{red}}^{k}\right]\right) \tag{4.14}$$

for $n, n' = 1, \ldots, N$ and $m = 1, \ldots, M$.

REMARK 4.3. *In the reduced basis scheme, we only use one collateral reduced basis space and one set of interpolation DOFs for both the operators $\mathcal{L}_E$ and $\mathcal{L}_I$. This is a feasible choice, whenever the operators implement similar "dynamics". Separate reduced basis spaces would include large redundancies in such scenarios. For a single collateral reduced basis, the* EI-ERRORESTIMATE *method simply returns the maximum interpolation error of operator evaluations for both operators.*

4.4. Offline/Online Decomposition. We now show that the reduced scheme from Definition 4.2 allows a full offline/online decomposition by summarizing the computed data fields and their theoretical complexity and size. The ability to precompute high-dimensional data in a single offline phase, is the key for efficient and fast online simulations.

The high-dimensional output during basis generation consists of

(i) the collateral reduced basis functions $\{\xi_m\}_{m=1}^{M} \subset \mathcal{W}_h$, corresponding interpolation DOFs Σ_M and the global index set $I_M := \bigcup \{I_\tau | \tau \in \Sigma_M\}$ for all operators subject to an empirical interpolation procedure, and

(ii) the reduced basis functions $\{\varphi_n\}_{n=1}^{N} \subset \mathcal{W}_h$.

For clarity of exposition, we describe the generated data only for one empirical interpolated operator $\mathcal{L}_h$. This is correct for schemes with purely implicit ($\mathcal{L}_h \equiv \mathcal{L}_I$) or purely explicit operator ($\mathcal{L}_h \equiv \mathcal{L}_E$) contributions, but can easily be extended to more complex situations. Before we proceed to reduced simulations, the high-dimensional data must be processed. Assuming that the initial data function is in a separable form, i.e.

$$u_0(\boldsymbol{\mu}) = \sum_{q=1}^{Q} \sigma_0^q(\boldsymbol{\mu}) u_0^q \tag{4.15}$$

with parameter dependent coefficient functions $\sigma_0^q : \mathcal{M} \to \mathbb{R}$ and parameter independent functions $u_0^q \in \mathcal{W}_h$ for $q = 1, \ldots, Q$, the parameter independent projections $\mathcal{P}_{\text{red}}\left[\mathcal{P}_h\left[u_0^q\right]\right]$ can be pre-computed with the already known reduced basis functions. In case of non–separable initial data, the initial data function can also be included into the collateral reduced basis generation process and be treated analogously to the discretization operators. Efficient evaluations of the operator during the online phase, depend on

(i) restrictions of the reduced basis functions to $\{\mathcal{R}_M\left[\varphi_n\right]\}_{n=1}^{N}$ with a restriction operator

$$\mathcal{R}_M : \mathcal{W}_h \to \mathcal{W}_h, \quad u_h = \sum_{i=1}^{H} \tau_i\left(u_h\right)\psi_i \mapsto \sum_{i \in I_M} \tau_i\left(u_h\right)\psi_i \tag{4.16}$$

and

(ii) the gram matrix $\mathbf{C}$ from (4.11), whose entries $C_{nm} = \langle \xi_m, \varphi_n \rangle$ depend on the *nodal* collateral reduced basis functions $\{\xi_m\}_{m=1}^M$ that need to be generated from the functions $\{q_m\}_{m=1}^M$ in a further pre-processing step. Note that this "basis transformation" is very efficient because of the special form of the collateral reduced basis (c.f. Remark 2.1).

REMARK 4.4. *In practice, discretization operators, like finite volume or finite element discretization operators are usually grid-based and the degrees of freedom correspond to distinctive points on the grid cells or its interfaces. In such case, a subgrid $\mathcal{S}_h \subset \mathcal{T}_h \subset \Omega$ as illustrated in Figure 4.1), might be necessary in order to compute the local operator evaluations and the restriction operator efficiently without an otherwise needed full grid traversal.*

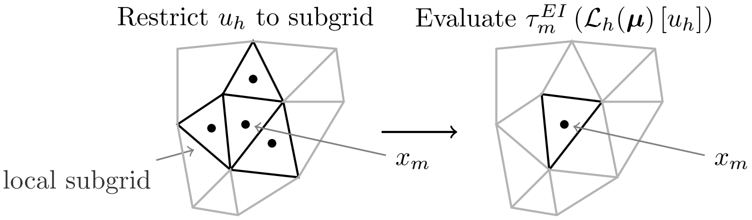

Restrict u_h to subgrid
Evaluate $\tau_m^{EI}\left(\mathcal{L}_h(\boldsymbol{\mu})\left[u_h\right]\right)$
local subgrid
x_m
x_m

FIGURE 4.1. *Illustration of finite volume operator evaluation on a local subgrid*

With the restricted reduced basis functions the local operator evaluations

$$\left(\tau_m^{EI} \circ \mathcal{L}_h(t^k, \boldsymbol{\mu})\right) \left[\sum_{n=1}^{N} a_n \varphi_n\right] = \left(\tau_m^{EI} \circ \mathcal{L}_h(t^k, \boldsymbol{\mu})\right) \left[\sum_{n=1}^{N} a_n \mathcal{R}_M \left[\varphi_n\right]\right] \tag{4.17}$$

have a complexity of $\mathcal{O}(N|I_M|) = \mathcal{O}(NM)$ for all $m = 1, \ldots, M$. This result can be applied to equations (4.13)-(4.14) and we see, that each of them lies in the complexity class $\mathcal{O}(NM^2)$. The generation of the Jacobian from equation (4.12) depends on $\mathcal{O}(N^2M^2)$ flops. This outreaches all other computations for the assembling of reduced matrices and vectors including matrix-matrix-multiplication of the reduced Jacobian with $\mathbf{C}$ consuming $\mathcal{O}(N^2M)$ flops. Therefore, one Newton step (4.7) of the reduced scheme has complexity $\mathcal{O}(N^2M^2 + N^3)$ including the costs for the linear equation solver. The computation of the Newton residual has costs $\mathcal{O}(NM^2)$. Unlike in the detailed simulation steps, the left hand side matrix in the linear equation system is not sparse. Because N is very small compared to the dimension of the detailed numerical scheme, we still expect the solution of the equation system to be much faster. We summarize that the reduced scheme is independent of the high dimensional data size H for each parameter after the offline-phase. A detailed comparison between costs for detailed and reduced simulations is given in Table 4.4.

5. A posteriori error estimation.

Rigorous a posteriori error estimators can be derived for the error between the reduced and detailed simulation. For a simple estimator with pure explicit discretization, we refer to [17]. Here, we derive a more accurate estimator which includes implicit discretization and residuals from Newton steps. It is assumed that a higher order empirical interpolation of the used operators

	detailed simulation	reduced simulation
Initial data projection	Eqn. (3.1): $\mathcal{O}(H)$	Eqn. (4.6): $\mathcal{O}(QN)$
Assembling of LHS and RHS in Newton step	Eqn. (3.3): $\mathcal{O}(H)$	Eqn. (4.7): $\mathcal{O}(N^2M^2)$
Solving Newton step	Eqn. (3.3): depending on linear solver, approximately $\mathcal{O}(H^2)$	Eqn. (4.7): $\mathcal{O}(N^3)$
Computing residual	Eqn. (3.5): $\mathcal{O}(H)$	Eqn. (4.8): $\mathcal{O}(NM^2)$

TABLE 4.1
Comparison of theoretical run–time complexities between detailed and reduced simulations.

is exact. Note, that this assumption is always fulfilled for $M + M' = H$ but for efficiency reasons in practice a much smaller value for M' needs to be used.

THEOREM 5.1. *Let $\left\{u_h^k(\boldsymbol{\mu})\right\}_{k=0}^K$ and $\left\{u_{\text{red}}^k(\boldsymbol{\mu})\right\}_{k=0}^K$ be solution trajectories obtained via the evolution schemes from Definitions 3.1 and 4.2 where the initial projection onto the reduced basis space is exact, i.e. $u_h^0(\boldsymbol{\mu}) \in \mathcal{W}_{\text{red}}$ for all $\boldsymbol{\mu} \in \mathcal{M}$. Further, we make two assumptions on the discretization operators $\text{Id} + \Delta t\mathcal{L}_I$ and $\text{Id} - \Delta t\mathcal{L}_E$. Firstly, the operators need to fulfill a lower respectively an upper Lipschitz continuity condition such that there exist constants $C_I, C_E > 0$, and for all $u, v \in \mathcal{W}_h$ the inequalities*

$$\left\|u - v + \Delta t\mathcal{L}_I\left[u\right] - \Delta t\mathcal{L}_I\left[v\right]\right\|_{\mathcal{W}_h} \geq \frac{1}{C_I}\left\|u - v\right\|_{\mathcal{W}_h} \tag{5.1}$$

$$\left\|u - v - \Delta t\mathcal{L}_E\left[u\right] + \Delta t\mathcal{L}_E\left[v\right]\right\|_{\mathcal{W}_h} \leq C_E\left\|u - v\right\|_{\mathcal{W}_h} \tag{5.2}$$

hold. Secondly, we assume the exactness of the empirical interpolation of the operators for a certain number of collateral reduced basis functions, i.e. there exists a positive integer $M' > 0$, such that

$$\mathcal{I}_{M+M'}\left[\mathcal{L}_I\right]\left[u_{\text{red}}^k(\boldsymbol{\mu})\right] = \mathcal{L}_I\left[u_{\text{red}}^k(\boldsymbol{\mu})\right] \quad \textit{and} \tag{5.3}$$

$$\mathcal{I}_{M+M'}\left[\mathcal{L}_E\right]\left[u_{\text{red}}^k(\boldsymbol{\mu})\right] = \mathcal{L}_E\left[u_{\text{red}}^k(\boldsymbol{\mu})\right] \tag{5.4}$$

for all $k = 0, \dots, K$ and $\boldsymbol{\mu} \in \mathcal{M}$.

Then, the norm of the error $e^k(\boldsymbol{\mu}) := u_h^k(\boldsymbol{\mu}) - u_{\text{red}}^k(\boldsymbol{\mu})$ can be bounded for $k = 0, \dots, K$ by $\eta_{N,M,M'}^k(\boldsymbol{\mu})$ which is an efficiently computable function defined by

$$\left\|e^k(\boldsymbol{\mu})\right\|_{\mathcal{W}_h} \leq \eta_{N,M,M'}^k(\boldsymbol{\mu}) := \\ \sum_{i=0}^{k-1} C_I^{k-i+1} C_E^{k-i}\left(\left\|\sum_{m=M+1}^{M+M'} \Delta t\theta_m^{i+1}(\boldsymbol{\mu})\xi_m\right\|_{\mathcal{W}_h} + \varepsilon^{New} + \left\|\Delta t R^{i+1}(\boldsymbol{\mu})\right\|_{\mathcal{W}_h}\right) \tag{5.5}$$

with a residual for the error due to the projection on the reduced basis space

$$\Delta t R^{k+1}(\boldsymbol{\mu}) := \left(\text{Id} + \Delta t\mathcal{I}_M\left[\mathcal{L}_I\right]\right)\left[u_{\text{red}}^{k+1}(\boldsymbol{\mu})\right] - \left(\text{Id} - \Delta t\mathcal{I}_M\left[\mathcal{L}_E\right]\right)\left[u_{\text{red}}^k(\boldsymbol{\mu})\right] \tag{5.6}$$

and empirical interpolation coefficients $\boldsymbol{\theta}^k(\boldsymbol{\mu}) := \left\{\theta_m^k(\boldsymbol{\mu})\right\}_{m=1}^{M+M'}$ defined by

$$\theta_m^k(\boldsymbol{\mu}) := \tau_m^{EI}\left(\mathcal{L}_I\left[u_{\text{red}}^k(\boldsymbol{\mu})\right] + \mathcal{L}_E\left[u_{\text{red}}^{k-1}(\boldsymbol{\mu})\right]\right). \tag{5.7}$$

Proof. For clarity of the exposition, we will discard all parameters $\boldsymbol{\mu}$ in this proof. First, we check that the residual norm $\left\|\Delta t R^k\right\|_{\mathcal{W}_h}$ can be computed efficiently, because with Definitions (5.6), (4.10)-(4.14) *and the empirical interpolation gram matrix* $\mathbf{X}$ *defined by*

$$(\mathbf{X})_{mm'} := \langle \xi_m, \xi_{m'} \rangle \tag{5.8}$$

it follows that

$$\begin{aligned}
\Delta t^2 \left\|R^{k+1}\right\|_{\mathcal{W}_h}^2 &= \left\langle \Delta t R^{k+1}, \Delta t R^{k+1} \right\rangle \\
&= \left(\mathbf{a}^{k+1} - \mathbf{a}^k\right)^T \mathbf{M} \left(\mathbf{a}^{k+1} - \mathbf{a}^k\right)^T \\
&\quad + 2\Delta t \left(\mathbf{l}_I\left[\mathbf{a}^{k+1}\right] + \mathbf{l}_E\left[\mathbf{a}^k\right]\right)^T \mathbf{C} \left(\mathbf{a}^{k+1} - \mathbf{a}^k\right) \\
&\quad + \Delta t^2 \left(\mathbf{l}_I\left[\mathbf{a}^{k+1}\right] + \mathbf{l}_E\left[\mathbf{a}^k\right]\right)^T \mathbf{X} \left(\mathbf{l}_I\left[\mathbf{a}^{k+1}\right] + \mathbf{l}_E\left[\mathbf{a}^k\right]\right).
\end{aligned}$$

Let us now derive the error bound. After each Newton iteration in the detailed numerical scheme, we obtain the equation

$$(\mathrm{Id} + \Delta t \mathcal{L}_I)\left[u_h^{k+1}\right] = (\mathrm{Id} - \Delta t \mathcal{L}_E)\left[u_h^k\right] + R_{h,New}^k \tag{5.9}$$

with Newton residual $\left\|R_{h,New}^k\right\|_{\mathcal{W}_h} \leq \varepsilon^{New}$.

The same can be obtained with (5.6) *for solutions of the reduced numerical scheme*

$$(\mathrm{Id} + \Delta t \mathcal{I}_M[\mathcal{L}_I])\left[u_{\mathrm{red}}^{k+1}\right] = (\mathrm{Id} - \Delta t \mathcal{I}_M[\mathcal{L}_E])\left[u_{\mathrm{red}}^k\right] + \Delta t R^{k+1}. \tag{5.10}$$

Subtracting (5.9) *from* (5.10) *leads to*

$$\begin{aligned}
&\underbrace{(\mathrm{Id} + \Delta t \mathcal{L}_I)\left[u_h^{k+1}\right] - (\mathrm{Id} + \Delta t \mathcal{I}_M[\mathcal{L}_I])\left[u_{\mathrm{red}}^{k+1}\right]}_{=:(I)} \\
&\qquad = \underbrace{(\mathrm{Id} - \Delta t \mathcal{L}_E)\left[u_h^k\right] - (\mathrm{Id} - \Delta t \mathcal{I}_M[\mathcal{L}_E])\left[u_{\mathrm{red}}^k\right]}_{=:(II)} \\
&\qquad\quad + R_{h,New}^{k+1} - \Delta t R^{k+1}.
\end{aligned} \tag{5.11}$$

After adding zeros to each of (I) and (II), these can be decomposed into terms that can (a) be estimated with the Lipschitz conditions and are (b) efficiently computable terms, only depending on low dimensional data

$$\begin{aligned}
(I) = &\underbrace{(\mathrm{Id} + \Delta t \mathcal{L}_I)\left[u_h^{k+1}\right] - (\mathrm{Id} + \Delta t \mathcal{L}_I)\left[u_{\mathrm{red}}^{k+1}\right]}_{=:(Ia)} \\
&+ \underbrace{(\mathrm{Id} + \Delta t \mathcal{L}_I)\left[u_{\mathrm{red}}^{k+1}\right] - (\mathrm{Id} + \Delta t \mathcal{I}_M[\mathcal{L}_I])\left[u_{\mathrm{red}}^{k+1}\right]}_{=:(Ib)},
\end{aligned} \tag{5.12}$$

$$\begin{aligned}
(II) = &\underbrace{(\mathrm{Id} - \Delta t \mathcal{L}_E)\left[u_h^k\right] - (\mathrm{Id} - \Delta t \mathcal{L}_E)\left[u_{\mathrm{red}}^k\right]}_{=:(IIa)} \\
&+ \underbrace{(\mathrm{Id} - \Delta t \mathcal{L}_E)\left[u_{\mathrm{red}}^k\right] - (\mathrm{Id} - \Delta t \mathcal{I}_M[\mathcal{L}_E])\left[u_{\mathrm{red}}^k\right]}_{=:(IIb)}.
\end{aligned} \tag{5.13}$$

Thereby, we have split the error propagation from the error at a previous time step $(Ia), (IIa)$ *from the error contribution through the empirical interpolation of the explicit and implicit discretization operators* $(Ib), (IIb)$*. Substituting the previous equations into* (5.11)*, bringing* (Ib) *on the right hand side and applying the Lipschitz condition* (5.1) *on it we obtain a bound for the error* $\left\|e^{k+1}\right\|_{\mathcal{W}_h}$ *by*

$$\begin{aligned}
\left\|e^{k+1}\right\|_{\mathcal{W}_h} &\leq C_I \left\|(\mathrm{Id}+\Delta t\mathcal{L}_I)\left[u_h^{k+1}\right] - (\mathrm{Id}+\Delta t\mathcal{L}_I)\left[u_{\mathrm{red}}^{k+1}\right]\right\|_{\mathcal{W}_h} \\
&= C_I \Bigg\| (\mathrm{Id}+\Delta t\mathcal{I}_M\left[\mathcal{L}_I\right])\left[u_{\mathrm{red}}^{k+1}\right] - (\mathrm{Id}+\Delta t\mathcal{L}_I)\left[u_{\mathrm{red}}^{k+1}\right] \\
&\qquad + (\mathrm{Id}-\Delta t\mathcal{L}_E)\left[u_h^{k}\right] - (\mathrm{Id}-\Delta t\mathcal{L}_E)\left[u_{\mathrm{red}}^{k}\right] \\
&\qquad + (\mathrm{Id}-\Delta t\mathcal{L}_E)\left[u_{\mathrm{red}}^{k}\right] - (\mathrm{Id}-\Delta t\mathcal{I}_M\left[\mathcal{L}_E\right])\left[u_{\mathrm{red}}^{k}\right] \\
&\qquad + R_{h,New}^{k+1} - \Delta t R^{k+1} \Bigg\|_{\mathcal{W}_h} \\
&\leq C_I \left(\left\| \sum_{m=M+1}^{M+M'} \Delta t \theta_m^{k+1} \xi_m \right\|_{\mathcal{W}_h} + C_E \left\|e^k\right\|_{\mathcal{W}_h} + \varepsilon^{New} + \left\|\Delta t R^{k+1}\right\|_{\mathcal{W}_h} \right),
\end{aligned} \tag{5.14}$$

where the last inequality uses the Lipschitz continuity (5.2) *of* $\mathcal{L}_E$*, the exactness assumptions* (5.3) *and* (5.4) *on* (Ib) *respectively* (IIb)*, the boundedness of the Newton residuals and the definition of the empirical interpolation coefficients. Resolving the recursion in* (5.14) *with initial error* $\left\|e^0\right\|_{\mathcal{W}_h} = 0$ *results in the proposed error bound.* □

Remark 5.2. *The Newton iteration error bound* ε^{New} *grows with the number of time instances. This is not a problem as the bound can be chosen arbitrarily small. However, for problems with exponential error growth in time* ($C_I > 1$) *it is reasonable to weigh the bound with the time steps size* Δt*.*

The error estimator (5.5) is similiar to the estimator from [21] for a finite element discretization of the viscous Burgers equation. There, it is proposed to adapt the operator constants C_E and C_I for each parameter by the so-called successive constraints method. This idea can also be transferred to the above described error estimator allowing for better effectivity bounds. A tight bound, especially for the implicit constant C_I is of great importance. If the bound is greater than one, the error estimator grows exponentially. Otherwise, it even ceases over time because individual snapshots can better approximate the detailed simulation because of the increasing smoothness of the solutions over time caused by diffusion.

It is obvious, that the error estimator respects an offline/online decomposition. A preliminary for the separation is the construction of a bigger collateral reduced basis $\boldsymbol{\xi}_{M+M'}$, but in the experimental section, we will observe that only few extra basis functions are needed for reasonable results. The evaluation of the estimator only includes low-dimensional terms or evaluations of the empirically interpolated operators $\mathbf{l}_I$ and $\mathbf{l}_E$. For a detailed discussion on the efficient evalution of these quantities, we refer to §4.4.

6. Experiments.

In this section, we demonstrate experiments for the presented reduced basis scheme. We consider two model examples that both fit into the setting of example (3.6)-(3.9). The first one is a Burgers problem with a purely implicit dis-

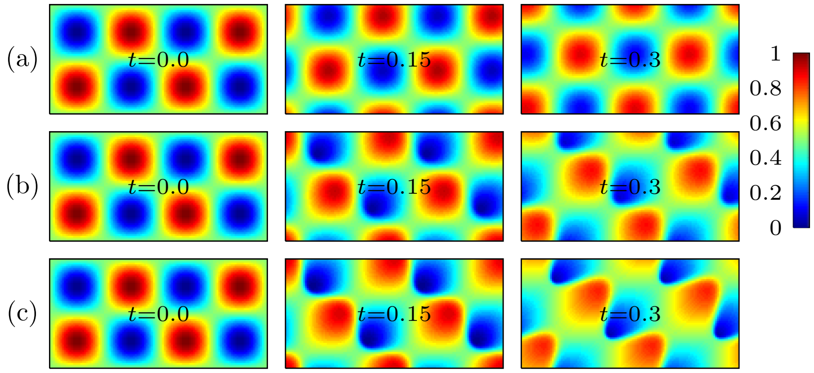


FIGURE 6.1. *Illustration of transport for smooth data. Snapshots at different time instants for (a) $\mu_1 = 1$ and (b) $\mu_1 = 1.5$ and (c) $\mu_1 = 2$.*

cretization. Preliminary results on this example without the new a posteriori error estimator and with explicit discretization have been presented in [15]. The second problem is based on a nonlinear non-stationary diffusion equation also with a purely implicit discretization. Both problems are nonlinear, but degenerate into linear ones for specific parameter configurations. The second example is also used for an evaluation of the a posteriori error estimator with special focus on its effectivity. The implementation is based on our MATLAB software package *RBmatlab* [26, 6].

6.1. Burgers equation. In a first example, we demonstrate the applicability to a nonlinear convection problem

$$\partial_t u - \nabla \cdot f(u) = 0 \tag{6.1}$$

with smooth initial data and a single parameter.

We choose $\Omega = [0, 2] \times [0, 1]$ with purely cyclical boundary conditions and fix the end time $T = 0.3$. We choose the nonlinear flux function $f(u; \boldsymbol{\mu}) := \mathbf{v}u^{\mu_1}$ with exponent μ_1 and space– and time–constant velocity field $\mathbf{v} = (1, 1)^T$, the initial data is a smooth function $u_0(x) = \frac{1}{2}(1 + \sin(2\pi x_1)\sin(2\pi x_2))$ for $x = (x_1, x_2)^T \in \Omega$. Overall, we consider the single parameter $\boldsymbol{\mu} = (\mu_1) \in \mathcal{M} := [1, 2]$ for the exponent in the flux of the evolution equation. We choose a 120×60 grid for decomposing Ω and $K = 100$ time–steps which satisfies the CFL-condition.

For this model setting, reduced basis spaces are generated with (A) a subsequent execution of EI-GREEDY and POD-GREEDY algorithms and (B) the PODEI-GREEDY algorithm. In the first case, the collateral reduced basis space is extended to its maximum size, i.e. until the reduction of the interpolation error stagnates due to machine precision and numerical constraints.

Figure 6.1 illustrates the time evolution of the solutions for different parameters, in particular the initial data which is independent of the parameter μ_1 and the final state for $\mu_1 = 1$ respectively $\mu_1 = 2$. The transition between linear convection ($\mu_1 = 1$) and the nonlinear non-viscous Burgers equation ($\mu_1 = 2$), can nicely be observed. In the latter case shock discontinuities emerge over time.

Offline phase. The reduced spaces are generated by (A) subsequent execution of the EI-GREEDY and POD-GREEDY algorithms and (B) the PODEI-GREEDY algorithm. All computations are taken on the PALMA cluster of the university of

Münster using 24 cores each running at 2.67 GHz. The implementation of the algorithms makes use of the simple parallelization technique mentioned in Remark 2.2. For the POD-GREEDY and the PODEI-GREEDY algorithms, we also apply the adaptation technique described in [14, 13], i.e. we begin with an initial uniform parameter set $\mathcal{M}_{\text{train}}^0 \subset \mathcal{M}$ and refine this set, every time the maximum of the error estimates over this set becomes small with respect to a randomly chosen validation set. In the experiments, we aim to assure a ratio of 1.1 between the maximum training set error and the validation set error. The initial parameter sampling set $\mathcal{M}_{\text{train}}^0$ for all three greedy algorithms consists of 26 uniformly distributed parameters in the parameter space interval $[1, 2]$.

In case (A) the generation of the collateral reduced basis with the EI-GREEDY algorithm takes 36 minutes and terminates with $M = 499$ basis functions after the tolerance of 10^{-10} for the interpolation error has been reached. The computation of the detailed simulations for all parameters takes only 100 seconds because of parallelization of these computations. The POD-GREEDY algorithm terminates after a bit less than 3 hours and produces a reduced basis space of dimension $N_{\max} = 249$. Note, that the main part of the offline run–time costs, namely the search for new basis functions, depends linearly on the number of available cores and therefore, the offline time can be controlled by using more processors. It is noteworthy, that the collateral reduced basis space is generated much faster than the reduced basis space. This is firstly because of the expensive POD step that needs to be computed in every POD-GREEDY extension step, and secondly, because the empirical interpolation errors are computed very fast after all necessary operator evaluations of detailed simulations have been cached.

In case (B) the initial collateral reduced basis is generated based on the coarse training sample $\mathcal{M}_{\text{train}}^{\text{coarse}} = \{1, 2\}$, and stops after three minutes with $M_{\text{small}} = 20$ generated basis functions. As the PODEI-GREEDY algorithm starts with a very small initial collateral reduced basis, the total offline time is smaller than in case (A) — especially for small reduced basis space dimensions. (c.f. Table 6.1) The algorithm stops after 3.27 hours. In both cases, the initial training sets $\mathcal{M}_{\text{train}}^0$ are not refined.

Next, we analyze the quality of the model order reduction induced by the reduced basis spaces. Figure 6.2(a) shows the growth of the Lebesgue constant defined in 2.2 with respect to the increasing collateral reduced basis size M. It can be clearly seen, that the increase is linear with a maximum Lebesgue value of 182. Figure 6.2(b) helps to understand how the empirical interpolation algorithm works. It illustrates the cell midpoints corresponding to the selected interpolation DOFs Σ_M and visualizes the selection order of the empirical interpolation algorithm by plotting points selected earlier in darker shades. It is visually comprehensible from the illustration that the algorithm realizes an obvious space compression, because it recognizes the space symmetry of the solution, such that the selected cell midpoints are all located in the lower left quarter of the domain. This means that interpolation DOFs for equivalent solution positions are weighted equally.

Online phase. In order to get a notion of the reduced simulations accuracy, in Figure 6.3 we illustrate the error convergence for the resulting reduced simulation scheme. We select a set $\mathcal{M}_{\text{test}} \subset \mathcal{M}$ of 100 random values for $\boldsymbol{\mu}$ not used during basis generation and determine the maximum error $\max_{\boldsymbol{\mu} \in \mathcal{M}_{\text{test}}} ||u_{\text{red}}(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})||_{L^\infty([0,T];\mathcal{W}_h)}$ between the reduced and the detailed simulations for different dimensionalities N and M. The resulting maximum error is plotted in logarithmic scale. The right hand side figure nicely shows, how a simultaneous increase of N and

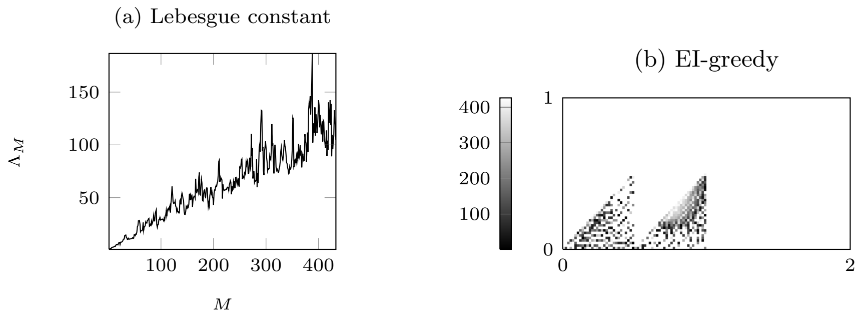


FIGURE 6.2. *Illustration of (a) growth of Lebesgue constant and (b) interpolation DOF selection for Burgers problem. DOFs corresponding to darker points are selected first.*

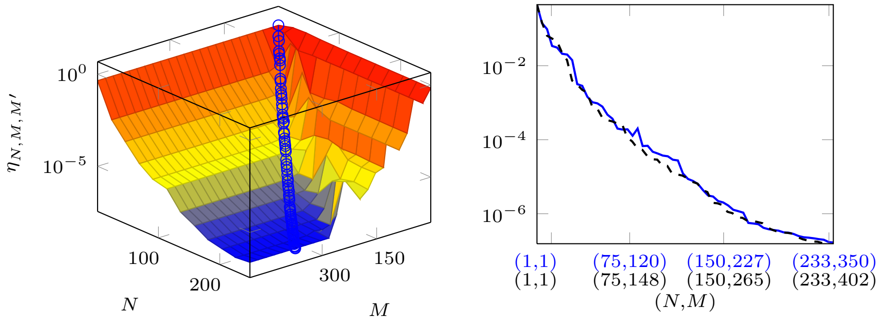


FIGURE 6.3. *Illustration of reduced basis error convergence for continuous initial data with varying dimensionalities N and M. The right hand side figure plots the error for simultaneously increased bases sizes N and M for the "optimal" ratio derived from the landscape and the error curve derived from the* PODEI-GREEDY *algorithm (dashed line).*

M reveals almost exponential convergence along a selected diagonal of the plot. This simultaneous increase is important: If M is fixed at a low value, increase of N over a certain limit can give an error increase induced by incorrectly approximated operator evaluations. If N is fixed, raising M gives no error improvement at some point.

The main goal of RB-methods is an accurate approximation under largely reduced simulation time. To assess these computation times, we determine the detailed and reduced simulation times over a sample of 100 random parameters and report the average run-times. These efficiency results are summarized in Tables 6.1(A)+(B) for different reduced basis sizes and for (A) subsequent generation of reduced basis and collateral reduced basis space and (B) the synchronised generation of both using the PODEI-GREEDY algorithm. In the first case the ratio between the dimensions for the empirical interpolation and the reduced basis are determined from the maximum basis sizes $499/249 \approx 2$ at which the algorithms stopped. Note, that this ratio cannot be assumed to be a good choice for smaller basis dimensions. In contrast, the M–N correlations in the second table are taken as inferred from the PODEI-GREEDY algorithm which sequentially expands both reduced basis spaces. It can be nicely observed, that this approach leads to better ratios, i.e. reaches smaller maximum

(A) EI-GREEDY + POD-GREEDY

N	M	ø-run–time[s]	max. error	offline time[h]
$H = 7200$	–	90.01	0.00	0
42	83	4.42	$1.15 \cdot 10^{-3}$	0.96
83	166	6.23	$6.03 \cdot 10^{-5}$	1.34
125	250	8.99	$7.43 \cdot 10^{-6}$	1.74
166	333	11.6	$8.33 \cdot 10^{-7}$	2.23
208	416	15.64	$2.47 \cdot 10^{-7}$	2.78
249	499	19.56	$2.38 \cdot 10^{-7}$	3.4

(B) PODEI-GREEDY

N	M	ø-run–time[s]	max. error	offline time[h]
$H = 7200$	–	90.01	0.00	0
42	72	4.44	$1.73 \cdot 10^{-3}$	0.54
83	144	6.04	$5.74 \cdot 10^{-5}$	1.09
125	216	8.37	$7.30 \cdot 10^{-6}$	1.55
167	288	11.92	$7.63 \cdot 10^{-7}$	2.08
208	360	15.08	$2.31 \cdot 10^{-7}$	2.69
233	402	16.48	$1.55 \cdot 10^{-7}$	3.27

Table 6.1

Run–time comparison for detailed simulation with reduced simulations of varying reduced dimensionalities. The average run–times and maximum errors are obtained over a test sample $\mathcal{M}_{test} \subset \mathcal{M}$*of size* 100*. The maximum error is obtained as* $\max_{(\boldsymbol{\mu},k)\in\mathcal{M}_{test}\times[0,\dots,K]} \left\|u_h^k(\boldsymbol{\mu}) - u_{\text{red}}^k(\boldsymbol{\mu})\right\|_{\mathcal{W}_h}$ *involving high dimensional error computations.*

errors with smaller basis spaces and therefore faster reduced simulation times. An exceptions is the second row of the tables with small dimensions $(N, M) = (42, 72)$. For reduced simulations with small reduced basis spaces, the PODEI-greedy suffers from the bad initial collateral reduced basis which needs a few extension steps to stabilize.

In general, however, the PODEI-greedy algorithm finds a very good choice for the M–N correlation. This observations is emphasized by the right hand side plot of Figure 6.3 comparing the maximum error decrease of the PODEI-greedy with an "optimal" error decrease curve manually derived from the neighbouring error landscape plot.

It can be seen nicely, that we obtain acceleration factors of $4.7 - 20$ depending on the dimensionalities of the reduced simulation. The acceleration factors obtained by the two different basis generation methods hardly differ.

6.2. Porous Medium Equation. In this section, we consider the porous medium equation given by the nonlinear diffusion problem

$$\partial_t u - m\Delta u^{\mu_1} = 0 \qquad \text{in } \Omega \times [0, T_{\max}], \tag{6.2}$$

$$u = c_0 + u_0 \qquad \text{on } \partial\Omega \times [0, T_{\max}], \tag{6.3}$$

$$u(\cdot, 0) = c_0 + u_0 \qquad \text{on } \Omega \times \{0\}\,, \tag{6.4}$$

on a rectangular domain $\Omega = [0, 1]^2$. The end time is fixed at $T_{\max} = 1.0$. The initial data function u_0 is a field of symmetrically arranged bar shaped concentrations illustrated in Figure 6.4(a). This gives us a non-smooth initial concentration depending on the initial parameter c_0.

The parameter vector is chosen as $\boldsymbol{\mu} = (\mu_1, m, c_0) \in \mathcal{M} := [1, 5] \times [0, 0.01] \times [0, 0.2]$ such that for $\mu_1 = 2$ we get the isothermal porous medium equation and for $\mu_1 > 2$

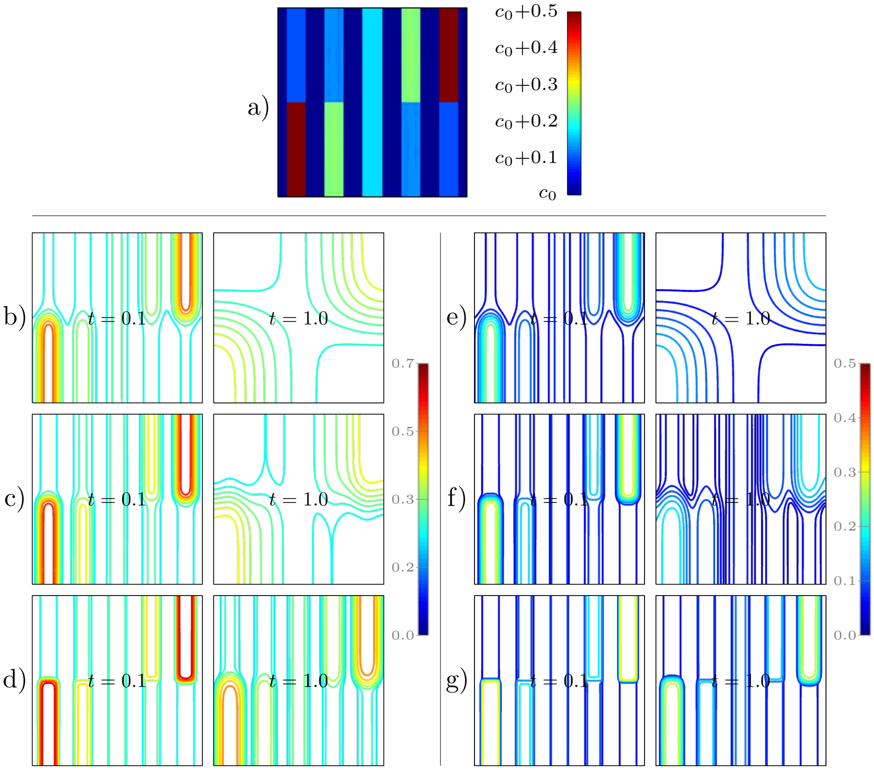
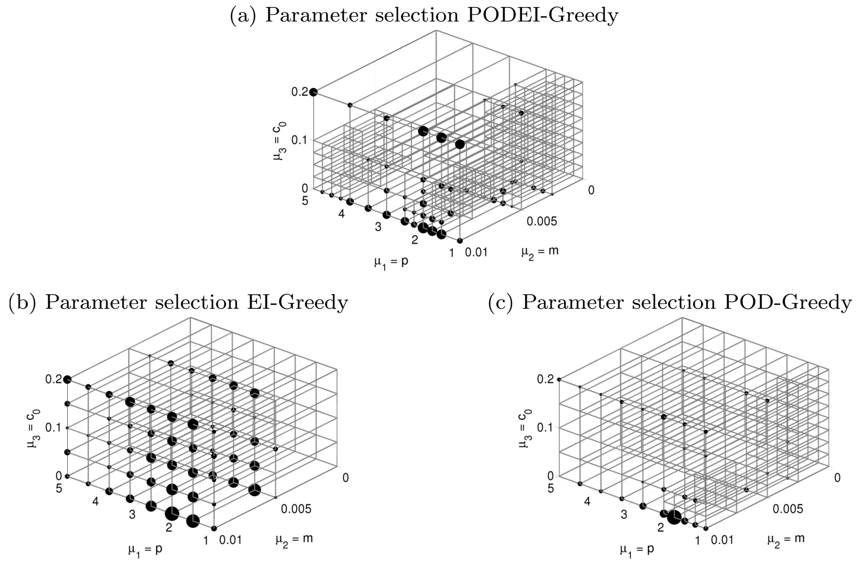


FIGURE 6.4. *Plot (a) depicts a color shading of the initial data. Below, isolines of reduced solutions are given at time instances $t = 0.1$ and $t = 1.0$ for different parameter vectors: (b) $\boldsymbol{\mu} = (1, 0.01, 0.2)$, (c) $\boldsymbol{\mu} = (2, 0.01, 0.2)$, (d) $\boldsymbol{\mu} = (4, 0.01, 0.2)$, (e) $\boldsymbol{\mu} = (1, 0.01, 0.0)$, (f) $\boldsymbol{\mu} = (2, 0.01, 0.0)$ and (g) $\boldsymbol{\mu} = (4, 0.01, 0.0)$.*

a porous medium equation with adiabatic flow. For $\mu_1 = 1$ it degenerates into the linear heat equation. Note also, that for c_0 close to zero, diffusion outside the bars is turned off in the non–linear case ($\mu_1 > 1$).

This effect can be observed in Figure 6.4(f)+(g) showing large diffusion effects inside the bars with high concentration after short time periods already, but almost none outside. Furthermore, Figure 6.4 clearly illustrates the nonlinear effects, as the diffusion is larger (more contour lines) in the bars with high initial concentration. An exception, of course, are the reduced solutions in the upper row modelling linear diffusion where the diffusivity stays the same globally. For the discretization, we chose again the finite volume scheme from §3.1 on a 100×100 grid for decomposing Ω and $K = 80$ time steps. The diffusivity is discretized implicitly, such that its nonlinearities are to be resolved with the Newton-Raphson method. If we make the reasonable assumption that the domain of the operator $\mathcal{L}_I$ as defined in Section §3.1 stays in the range $[0, 1]$, the operator fulfills the Lipschitz condition (5.1) with $C_I = 1$.

Offline phase. Like in the previous example, we compute the reduced basis spaces on the PALMA cluster using 24 cores for (A) subsequent execution of EI-GREEDY and POD-GREEDY algorithms and (B) the PODEI-GREEDY algorithm. This time a third run (C) with a "true" error indicator $\eta(\boldsymbol{\mu}) = \left\| u_h^K(\boldsymbol{\mu}) - u_{\text{red}}^K(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$


(a) Parameter selection PODEI-Greedy

(b) Parameter selection EI-Greedy

(c) Parameter selection POD-Greedy

FIGURE 6.5. *Illustration of the final training parameter sets $\mathcal{M}_{train}^{m_{ref}}$ after (a) $m_{ref} = 3$, (b) $m_{ref} = 0$ and (c) $m_{ref} = 1$ refinement steps. The vertices match with the parameters in the training sets and the overlaying bubble plots illustrates how frequently a parameter is picked for basis extension.*

instead of the error estimator is executed in order to assess the suitability of the error estimator for the basis generation. This question is discussed in §6.3 below.

Again, we apply the adaptation technique described in [13] on the parameter space. The fixed parameter sampling set for the EI-GREEDY algorithm consists of 120 parameter vectors distributed as illustrated in Figure 6.5(b). The vertices of the drawn grid match with the training parameters. For the POD-GREEDY algorithm the same initial parameter set $\mathcal{M}_{\text{train}}^{0}$ is chosen, whereas the PODEI-GREEDY algorithm starts with 30 training parameters. The result of the adaptive refinement procedures is illustrated in Figures 6.5(a)+(c). The training set in case (A) has been refined once with a final number of 209 parameter vectors, and in case (B) where we started with a coarser grid, has been refined three times resulting in a set of size 305. Furthermore, Figure 6.5 shows that parameters which are often selected for basis extension, correlate with the refined parts of the grids. Here, we nicely observe two facts: First, solutions that show a complex evolution over time, are selected more frequently until they are approximated well enough, and second for parameters with solutions evolving more linear in time, few or even zero snapshots are sufficient, because these can be approximated by linear combinations of other basis functions.

In case (A) the empirical interpolation algorithm takes 38.5 minutes until it reaches the final number of 425 basis functions and the computation of the detailed simulations for all training set parameters takes 4 minutes. The reduced basis space generation terminates after 2.1 hours and 99 generated reduced basis functions. It does not reach the targeted error of $\varepsilon_{\text{tol}} = 10^{-4}$ because after 99 basis extensions no snapshots can be found which reduce the maximum error estimate over the training parameter set. Figure 6.6(b) illustrating the decrease of the error estimates during the basis extension suggest that the reason for the stagnation comes from a bad estimation

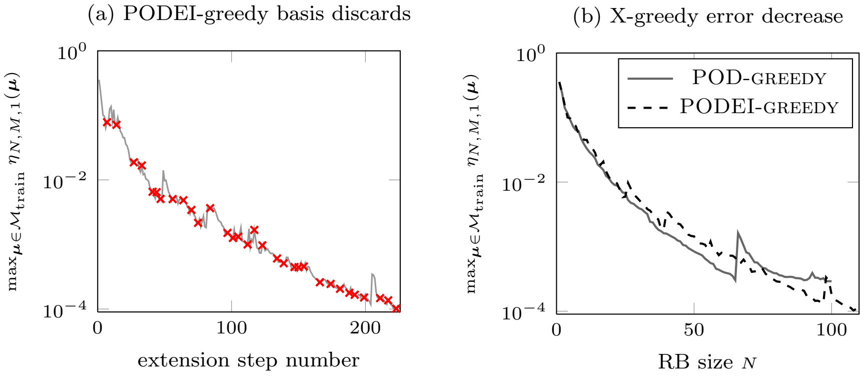


FIGURE 6.6. *Illustration of (a) the extension steps during* PODEI-GREEDY *at which reduced basis functions were discarded (marked with a cross) and (b) the error decrease during basis extension with growing reduced basis size.*

of the empirical operator interpolation: The stagnation begins after the refinement step which is indicated by the strong peak in the error curve, i.e. when the current training set differs from the one used for the EI-GREEDY algorithm.

In case (B) the synchronised generation of basis functions takes about 4 hours, but reaches the a priori given error tolerance $\varepsilon_{\text{tol}} = 10^{-4}$. The initial collateral reduced basis is generated in about two minutes with a coarse training set $\mathcal{M}_{\text{train}}^{\text{coarse}}$ containing only the extremal points of the parameter space and after $M_{\text{small}} = 20$ basis functions have been generated.

The Lebesgue constant Λ_M of the collateral reduced basis space (c.f. 2.2 grows in both cases linearly with a maximum of 153 in case (A) and 203 in case (B).

For details, on the offline computation times, we refer to Table 6.2. The table nicely shows that for case (B) very reasonable M–N correlations are inferred from the synchronised basis extension and even a basis of better quality is produced: Comparing lines with similar maximum errors over the validation set of 100 parameters, the reduced basis from (B) needs less basis vectors which have been generated in a shorter offline phase. For example the maximum error of $3.54 \cdot 10^{-5}$ is reached with basis dimensions $(N, M) = (93, 358)$ which can be generated in 2.72 hours, whereas Table 6.2(A) shows that the worse error of $4.06 \cdot 10^{-5}$ needs basis size $(N, M) = (99, 425)$ and basis generation time of 3.3 hours.

Figure 6.6(a) illustrates how often and how frequently basis functions are discarded during the execution of the PODEI-GREEDY algorithm. The right hand side plot compares the error estimation decrease during POD-GREEDY and the PODEI-GREEDY extension. The latter needs more extension steps in order to reach a certain basis space dimension, because some basis functions are discarded as indicated by the crossed marks. Note also that both error curves have intermediate peaks because of the adaptation of the parameter training set. As the maximum error is computed over the current training parameter set, it grows after such a refinement step, when more parameters are added.

Online phase. Figure 6.7 shows cross-section plots of detailed and reduced simulation snapshots of the two worst solutions from a set of 100 randomly chosen solution

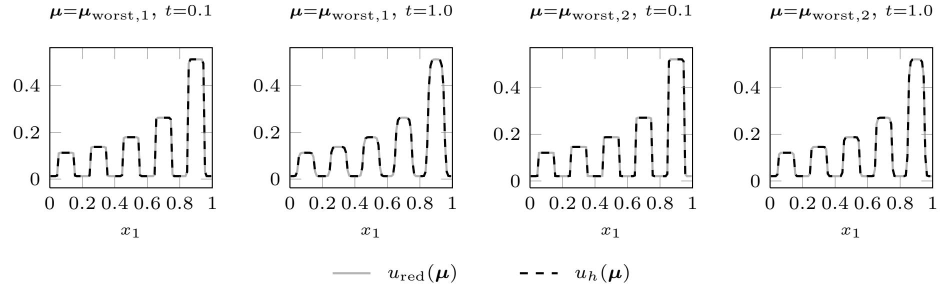


FIGURE 6.7. *Comparison of cross–section plots at* $x_2 = 0.6$ *between detailed and reduced simulation snapshots for different time instances and parameters with worst and seconds worst error from a random test set of* 100 *parameters:* $\mu_{\mathit{worst,1}} = (1.349, 7.293 \cdot 10^{-5}, 0.013)$ *and* $\mu_{\mathit{worst,2}} = (1.737, 5.758 \cdot 10^{-5}, 0.021)$.

trajectories. Visually, there are no differences between the dashed curve of the detailed and the solid curve of the reduced solution snapshots. This indicates that dispersion effects arising from the additional approximation are negligible for this example.

To quantify the reduced simulations quality, we proceed exactly as we did in the previous section for the Burgers problem. We again pick a test sample $\mathcal{M}_{\text{test}} \subset \mathcal{M}$ of 100 randomly chosen values from the parameter space, measure the error $\|u_h(\mu) - u_{\text{red}}(\mu)\|_{L^\infty([0,T_{\max}];\mathcal{W}_h)}$ for all $\mu \in \mathcal{M}_{\text{test}}$ and compare the computation times of detailed and reduced simulations. Results for different magnitudes of the reduced bases dimensions M and N are shown in Table 6.2. Note, that the run–times are averaged over the test parameter set, and actually show a high deviation from this mean by factors up to 10, because the number of Newton steps that are needed to proceed between time–steps varies noticeably. For linear problems one Newton step is enough, whereas up to a maximum of 30 Newton steps for stronger non–linearities, i.e. $\mu_2 > 1$ are necessary. Consequently, the acceleration factors for reduced simulations with maximum reduced basis dimensions also differ notably.

6.3. A posteriori error estimator. The a posteriori error estimator from §5 has two main purposes: It should first give a tight and rigorous bound on the real error made through the model order reduction, and second improve the run–time of basis generation algorithms by providing an efficient and trust–worthy error indicator. In this section, we evaluate how good both these tasks are fulfilled.

In a first test, we check the efficiency of the error estimator

$$\lambda(\mu) := \frac{\eta^K_{N,M,M'}(\mu)}{\left\|u^K_h(\mu) - u^K_{\text{red}}(\mu)\right\|_{\mathcal{W}_h}} \tag{6.5}$$

for a random sample of 100 parameters and different values for the extra collateral basis functions M' used to estimate the interpolation error. We expect $\lambda(\mu)$ to be greater than one, meaning the estimator is rigorous, i.e. does not underestimate the real error. On the other hand, it is desirable that the efficiency is very close to one.

Recall our assumption, that a large enough collateral reduced basis allows interpolated operator evaluations to be almost exact. This gives rise to assess the empirical operator interpolation error with basis dimension M by comparing it to the finer interpolation with basis dimension $M + M'$. One question that needs to be answered

(A) EI-greedy + POD-greedy, $\eta_{N,M,1}$

N	M	ø-run–time[s]	max. error	offline time[h]
$H = 10000$	–	55.38	0.00	0
17	71	1.57	$3.56 \cdot 10^{-3}$	1.35
33	142	1.95	$8.33 \cdot 10^{-4}$	1.67
50	213	2.51	$2.08 \cdot 10^{-4}$	2.07
66	283	3.19	$5.88 \cdot 10^{-5}$	2.43
83	354	4.07	$5.55 \cdot 10^{-5}$	2.88
99	425	5.3	$4.06 \cdot 10^{-5}$	3.3

(B) PODEI-greedy, $\eta_{N,M,1}$

N	M	ø-run–time[s]	max. error	offline time[h]
$H = 10000$	–	55.38	0.00	0
19	72	1.61	$3.01 \cdot 10^{-3}$	0.16
37	143	2.07	$7.90 \cdot 10^{-4}$	0.45
56	215	2.67	$1.66 \cdot 10^{-4}$	1.01
74	286	3.6	$6.36 \cdot 10^{-5}$	1.69
93	358	4.83	$3.54 \cdot 10^{-5}$	2.72
111	429	6.55	$1.96 \cdot 10^{-5}$	4.02

(C) EI-greedy + POD-greedy, $\eta = \|\cdot\|$

N	M	ø-run–time[s]	max. error	offline time[h]
$H = 10000$	–	55.38	0.00	0
17	71	1.46	$2.97 \cdot 10^{-3}$	1.26
33	142	1.83	$5.87 \cdot 10^{-4}$	1.91
50	213	2.31	$1.24 \cdot 10^{-4}$	2.64
67	283	3.32	$6.30 \cdot 10^{-5}$	3.49
83	354	4.06	$3.19 \cdot 10^{-5}$	4.31
100	425	5.99	$1.34 \cdot 10^{-5}$	5.64

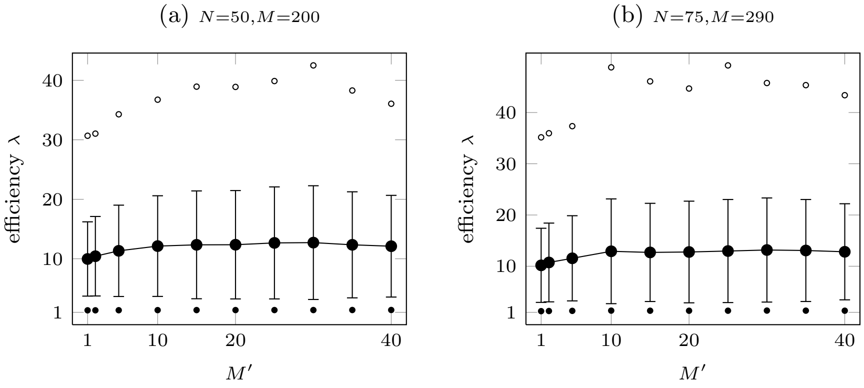
Table 6.2

Run–time comparison for detailed simulation with reduced simulations of varying reduced dimensionalities. The average run–times and maximum errors are obtained over a test sample $\mathcal{M}_{test} \subset$ $\mathcal{M}$of size 100. The maximum error is obtained as $\max_{(\boldsymbol{\mu},k)\in\mathcal{M}_{test}\times[0,\ldots,K]} \left\| u_h^k(\boldsymbol{\mu}) - u_{\mathrm{red}}^k(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$. Subtable (c) is based on a reduced basis generated with a "true" error indicator $\eta(\boldsymbol{\mu}) = \left\| u_h^K(\boldsymbol{\mu}) - u_{\mathrm{red}}^K(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$.

empirically here, is whether this assumption is valid and if yes, how big M' needs to be chosen. The results of our experiments are illustrated in Figure 6.8: The plot shows statistical data of the measured effectivities for different error estimators, i.e. different values for M'. We observe that the mean effectivity is slightly above 10 for $M' = 1$ and stabilizes at about 12 for small M' already. The latter gives rise to our assumption that the empirical interpolation error is well approximated by a small set of extra basis functions.

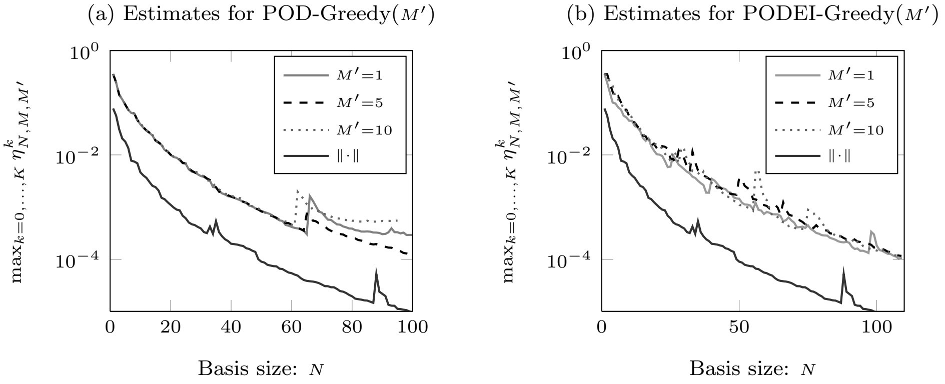
As the standard deviation of the error estimator's efficiency is still in a reasonable range for the sample parameters in this test, we can expect the estimator to have a good qualification as an error indicator for the POD-greedy and the PODEI-greedy algorithms. This is confirmed in further test runs where both algorithms are run several times with different choices of M' in the error indicator for the greedy search. The result are shown in Figure 6.9.

The plots show the maximum error estimates for all parameters from the training set at each reduced basis extension step during the greedy search algorithm. Here, the lower black line corresponds to the error curve of the reference run (C) where the


(a) $N{=}50,M{=}200$

(b) $N{=}75,M{=}290$

FIGURE 6.8. *Error bar plot showing mean and standard deviation of error estimator efficiency over a sample of* 100 *random parameters for different values of* M'. *The dots indicate the minimum and maximum efficiency.*


(a) Estimates for POD-Greedy(M')

(b) Estimates for PODEI-Greedy(M')

FIGURE 6.9. *Comparison of estimated error decrease during basis generation with (a)* POD-GREEDY *and (b)* PODEI-GREEDY *algorithm. Different error indicators are used in order to select the worst approximated trajectories (c.f. Algorithm 4.1). The error indicators vary in the number of collateral reduced basis functions* M' *used in order to approximate the empirical interpolation error. The lower indicator curves depict the error decrease during a* POD-GREEDY *with "true" error indicator* $\eta(\boldsymbol{\mu}) = \left\| u^K_{\mathrm{red}}(\boldsymbol{\mu}) - u^K_h(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$.

greedy extension algorithm is used with the "true" error $\left\| u^K_{\mathrm{red}}(\boldsymbol{\mu}) - u^K_h(\boldsymbol{\mu}) \right\|_{\mathcal{W}_h}$ as an indicator. We observe, that, in general, all plots show an error decrease at a rate similar to the reference plot. Only the POD-GREEDY algorithm makes an exception — after the last adaptation of the training parameter set at about $N = 60$. The reason for this behaviour can be the non–adaptive training parameter set used for the collateral reduced basis space. No matter what value has been chosen for M', the runs show no qualitiative deviation. Table 6.2 shows that the error reduction obtained with reduced basis spaces generated with greedy search algorithms based on the error estimator is of comparable quality to the "optimal" values of Table 6.2(C), while the offline time is reduced significantly. The effect is especially salient for small basis spaces generated with the PODEI-GREEDY algorithm. All this confirms our assumption that the estimator is a valid error indicator for the greedy search — already with a small number of extra collateral basis functions.

7. Outlook and conclusion.

With sophisticated parametrized evolution problems in mind, we developed an empirical operator interpolation technique for nonlinear discrete operators and their directional derivatives. This allows to apply the reduced basis framework to nonlinear evolution problems. We exemplified the approach for finite volume approximations where Newton's method is used to solve the resulting nonlinear systems. We derived a new a posteriori error estimator for the error between reduced and detailed simulations, and showed that it helps to accelerate the reduced basis generation during the offline phase. Furthermore, we introduced a new greedy algorithm that extends the reduced basis spaces for the empirical interpolation of the discrete operators and the numerical scheme in a synchronised way. In particular the relation of the dimensions of the reduced and the collateral reduced basis spaces is detected automatically. This new algorithm avoids high dimensional computations for large training parameter sets in the offline phase for the empirical interpolation and frees the experimenter from the necessity to make a priori assumptions on the M–N correlation. In contrast, the subsequent execution of EI-GREEDY and POD-GREEDY algorithms relies on a good initial parameter space in order to minimize the pre–computed simulations and on an assumption on the relation between the interpolation error of the EI-GREEDY algorithm and the targeted error of the POD-GREEDY algorithm.

We demonstrated, that the reduced basis methodology respectively the interpolation procedure is able to detect spatial redundancy. In the given examples, it realizes not only spatial compression but even symmetry detection. In our experiments, the reduced models have shown an acceleration of at least one order of magnitude. The reduced basis framework developed in this paper, allows to consider numerical schemes with all kind of implicit and explicit discretizations.

For our problems, the reduced basis sizes grow very fast and in case of even more complex dependencies of the solutions on the parameter, the model order reduction effects could become void with the current procedure. As future work will deal with more complex problems depending on the coupling of different equations or very high dimensional parametrizations, more sophisticated basis generation algorithms are necessary. Here, we can improve the methods by more intelligent search algorithms in the parameter space and by producing smaller bases for reasonably selected subsets of the parameter space or the time interval as proposed in [8, 13].

Acknowledgement. The authors would like to acknowledge the reviewers for their numerous and helpful comments on this work.

REFERENCES

[1] M. Barrault, Y. Maday, N. Nguyen, and A. Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris Series I, 339 (2004), pp. 667–672.

[2] C. Canuto, T. Tonn, and K. Urban, *A-posteriori error analysis of the reduced basis method for non-affine parameterized nonlinear pde's*, SIAM J. Numer. Anal., 47 (2009), pp. 2001–2022.

[3] J. Carrillo, *Entropy solutions for nonlinear degenerate problems*, Arch. Ration. Mech. Anal., 147 (1999), pp. 269–361.

[4] S. Chaturantabut and D. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764.

[5] P. Ciarlet, *The finite element method for elliptic problems*, North-Holland, 1978.

[6] M. Drohmann, B. Haasdonk, S. Kaulmann, and M. Ohlberger, *A software framework for reduced basis methods using DUNE-RB and RBmatlab*, in Advances in DUNE, A. Dedner, B. Flemisch, and R. Klöfkorn, eds., Springer, to appear.

[7] M. Drohmann, B. Haasdonk, and M. Ohlberger, *Reduced basis method for finite volume approximation of evolution equations on parametrized geometries*, in Proceedings of ALGORITMY 2009, 2008, pp. 111–120.

[8] J. L. Eftang, D. J. Knezevic, and A. T. Patera, *An hp certified reduced basis method for parametrized parabolic partial differential equations*, Mathematical and Computer Modelling of Dynamical Systems, 17 (2011), pp. 395–422.

[9] M. Grepl, *Reduced-basis Approximations and a Posteriori Error Estimation for Parabolic Partial Differential Equations*, PhD thesis, Massachusetts Institute of Technology, May 2005.

[10] M. Grepl, Y. Maday, N. Nguyen, and A. Patera, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, M2AN, Math. Model. Numer. Anal., 41 (2007), pp. 575–605.

[11] M. Grepl and A. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations*, M2AN, Math. Model. Numer. Anal., 39 (2005), pp. 157–181.

[12] B. Haasdonk, *Convergence rates of the POD-greedy method*, Tech. Rep. 23, SimTech Preprint 2011, University of Stuttgart, 2011.

[13] B. Haasdonk, M. Dihlmann, and M. Ohlberger, *A training set and multiple bases generation approach for parametrized model reduction based on adaptive grids in parameter space*, Mathematical and Computer Modelling of Dynamical Systems, 17 (2011), pp. 423–442.

[14] B. Haasdonk and M. Ohlberger, *Adaptive basis enrichment for the reduced basis method applied to finite volume schemes*, in Proc. 5th International Symposium on Finite Volumes for Complex Applications, 2008, pp. 471–478.

[15] B. Haasdonk and M. Ohlberger, *Reduced basis method for explicit finite volume approximations of nonlinear conservation laws*, in Proc. 12th International Conference on Hyperbolic Problems: Theory, Numerics, Application, 2008.

[16] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, M2AN, Math. Model. Numer. Anal., 42 (2008), pp. 277–302.

[17] B. Haasdonk, M. Ohlberger, and G. Rozza, *A reduced basis method for evolution schemes with parameter-dependent explicit operators*, Electron. Trans. Numer. Anal., 32 (2008), pp. 145–161.

[18] K. Karlsen and N. Risebro, *On the uniqueness and stability of entropy solutions of nonlinear degenerate parabolic equations with rough coefficients*, Discrete Contin. Dyn. Syst., 9 (2003), pp. 1081–1104.

[19] D. Kröner, *Numerical Schemes for Conservation Laws*, John Wiley & Sons and Teubner, 1997.

[20] Y. Maday, *Reduced basis method for the rapid and reliable solution of partial differential equations*, in Proceedings of International Conference of Mathematicians, E. M. Society, ed., 2006, pp. 1–17.

[21] N. C. Nguyen, G. Rozza, and A. Patera, *Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers' equation*, Calcolo, 46 (2009), pp. 157–185.

[22] A. Patera and G. Rozza, *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*, MIT, 2007. Version 1.0, Copyright MIT 2006-2007, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering.

[23] G. Rozza, *Shape design by optimal flow control and reduced basis techniques: Applications to bypass configurations in haemodynamics*, PhD thesis, École Polytechnique Fédérale de Lausanne, November 2005.

[24] T. Tonn, *Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs*, PhD thesis, Universität Ulm, 2011.

[25] K. Veroy, C. Prud'homme, and A. Patera, *Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds*, C. R. Math. Acad. Sci. Paris Series I, 337 (2003), pp. 619–624.

[26] http://morepas.org/software/.